



**Yale University**  
**Department of Computer Science**

**On Stable Route Selection for Interdomain Traffic  
Engineering: Models and Analysis**

Hao Wang<sup>1</sup>      Haiyong Xie<sup>2</sup>      Yang Richard Yang<sup>3</sup>  
Li (Erran) Li<sup>4</sup>      Yanbin Liu<sup>5</sup>      Avi Silberschatz<sup>6</sup>

YALEU/DCS/TR-1316

February 8, 2005

<sup>1</sup>Computer Science Department, Yale University, New Haven, CT 06520-8285, USA. Email: hao.wang@yale.edu. Supported by NSF grant ANI-0207399.

<sup>2</sup>Computer Science Department, Yale University, New Haven, CT 06520-8285, USA. Email: yong@cs.yale.edu. Supported by NSF grant ANI-0238038.

<sup>3</sup>Computer Science Department, Yale University, New Haven, CT 06520-8285, USA. Email: yry@cs.yale.edu. Supported in part by NSF grants ANI-0207399 and ANI-0238038.

<sup>4</sup>Networking Research Lab, Bell Labs, Lucent, Holmdel, NJ 07733-3030 Email: li.li@bell-labs.com.

<sup>5</sup>Department of Computer Sciences, The University of Texas at Austin, Austin, TX 78712, USA. Email: ybliu@cs.utexas.edu.

<sup>6</sup>Computer Science Department, Yale University, New Haven, CT 06520-8285, USA. Email:



# On Stable Route Selection for Interdomain Traffic Engineering: Models and Analysis

Hao Wang\*   Haiyong Xie<sup>†</sup>   Yang Richard Yang<sup>‡</sup>   Li (Erran) Li<sup>§</sup>  
Yanbin Liu<sup>¶</sup>   Avi Silberschatz<sup>||</sup>

## Abstract

BGP route selection is increasingly being used by ASes to achieve interdomain traffic engineering objectives. One fundamental feature of route selection for interdomain traffic engineering is that routes for a set of destinations be chosen jointly to satisfy traffic engineering constraints and meet traffic engineering objectives. In this paper, we present a general model of route selection for interdomain traffic engineering by allowing the routing of multiple destinations to be coordinated. We identify potential routing instability and inefficiency by showing that there exist networks where the interaction of the route selection of multiple destinations can cause routing instability, even though the networks are guaranteed to converge to a unique route selection when each destination is considered alone. We derive a sufficient condition to guarantee routing convergence. We also show that the constraints on local policies imposed by business considerations in the Internet can guarantee stability without global coordination. Using realistic Internet topology, we evaluate the extent to which routing instability of interdomain traffic engineering can happen when the constraints are violated. We further generalize the preceding model under two extensions. One, we investigate the general model that the preference of an AS depends on not only its egress routes to the destinations but also its inbound traffic pattern. Second, instead of studying a specific route selection algorithm, we study a general class of route selection algorithms which we call *rational route selection algorithms*. We present a sufficient condition to guarantee routing convergence in a heterogeneous network where each AS runs any rational route selection algorithm. We also show that there

---

\*Computer Science Department, Yale University, New Haven, CT 06520-8285, USA. Email: hao.wang@yale.edu. Supported by NSF grant ANI-0207399.

<sup>†</sup>Computer Science Department, Yale University, New Haven, CT 06520-8285, USA. Email: yong@cs.yale.edu. Supported by NSF grant ANI-0238038.

<sup>‡</sup>Computer Science Department, Yale University, New Haven, CT 06520-8285, USA. Email: yry@cs.yale.edu. Supported in part by NSF grants ANI-0207399 and ANI-0238038.

<sup>§</sup>Networking Research Lab, Bell Labs, Lucent, Holmdel, NJ 07733-3030 Email: li.li@bell-labs.com.

<sup>¶</sup>Department of Computer Sciences, The University of Texas at Austin, Austin, TX 78712, USA. Email: ybliu@cs.utexas.edu.

<sup>||</sup>Computer Science Department, Yale University, New Haven, CT 06520-8285, USA. Email: avi@cs.yale.edu. Supported in part by NSF.

exist networks which will have persistent route oscillations even when the ASes strictly follow the constraints imposed by business considerations, and adopt *any* rational route selection algorithms.

## 1 Introduction

The global Internet consists of a large number of interconnected autonomous systems (AS), where each AS (*e.g.*, AT&T) is administrated autonomously. Recently, ASes are increasingly adopting local route selection policies to achieve their interdomain traffic engineering objectives (*e.g.*, [45]). We have recently conducted an email survey of ISPs, and the results indicate that many ISPs choose routes to achieve their interdomain traffic engineering objectives, such as satisfying the capacity constraints of links between neighboring ASes (*e.g.*, [5]), load-balancing interdomain traffic, and/or minimizing cost (*e.g.*, [26]).

Despite this emerging trend, so far there are few systematic studies on the stability and efficiency of the global Internet with route selection for interdomain traffic engineering. As several researchers pointed out [8,45]: “the state of the art for interdomain traffic engineering is extremely primitive.” Learning anecdotal incidents causing instability in the Internet (*e.g.*, [39]) and recognizing the potential issues of using route selection for interdomain traffic engineering, researchers have proposed both configuration guidelines (*e.g.*, [8,45]) and alternatives/extensions to the current interdomain routing protocol (*e.g.*, [1,39,49]). However, since the essential features of route selection for interdomain traffic engineering have not been pinpointed and analyzed [10], it is unclear whether these guidelines and new protocols can produce stable and efficient route selections in the global Internet.

A major breakthrough was made recently when Griffin *et al.* [22, 28, 29, 32, 47] proposed systematic models to study the stability of path-vector interdomain routing. In particular, these previous models identified the existence of policy disputes as a potential reason for routing instability. By routing instability, they mean persistent route oscillations even though the network topology is stable.

Although these previous models can already capture a wide range of potential route selection behaviors for interdomain traffic engineering, since they require that the routing decisions of different destinations be separated, they cannot be applied to study a large class of common traffic engineering behaviors. In particular, a fundamental feature of route selection for interdomain traffic engineering in particular and traffic engineering in general is that route selection constraints (*e.g.*, traffic assigned to a link is within link capacity) and/or objective functions (*e.g.*, load balance) involve the route selection of multiple destinations. Thus, in route selection for interdomain traffic engineering, whether a route will be chosen by an AS for a given destination will depend on what routes are available or chosen for other destinations. For example, if an AS selects routes for each destination independently without considering the chosen/available routes of other destinations, in the worst case it may choose the same access link for all destinations, violating link capacity constraints and/or causing load imbalance. By requiring that the routing of each destination be separated, the previous models apply only to a network where there is no AS whose routing policies require it to coordinate its route selection to multiple destinations.

In this paper, we first identify that there exist networks where the coordination of the route selection of multiple destinations due to interdomain traffic engineering considerations can cause routing instability, even though the networks are guaranteed to converge when each destination is considered alone. The identification of such routing instability shows that a general route selection model is needed to analyze the stability of route selection for interdomain traffic engineering. Motivated by the need, we propose a route selection model where each AS partitions the destinations into arbitrary subsets, and for each subset, the AS can coordinate the route selection of the destinations in the subset. This model is very general and is the first general model which captures the essence of route selection for interdomain traffic engineering.

Using the model, we analyze the stability of path-vector interdomain routing when ASes choose egress routes to achieve interdomain traffic engineering objectives. We call this problem *the stable route selection for egress interdomain traffic engineering problem*. We propose the construction of *P-graphs*, and derive sufficient conditions based on the properties of *P-graphs* to guarantee the convergence of route selections under interdomain traffic engineering.

We also investigate the efficiency of route selection for interdomain traffic engineering. We show an example with multiple stable route selections but one of them is not Pareto optimal. These results clearly demonstrate the intrinsic challenges of route selection for interdomain traffic engineering in a generic network. It will be challenging to achieve stable and efficient outcomes for general networks even when ASes adopt explicit negotiations.

The route selection of Internet has its own special properties. Applying our general results, we investigate whether route selection for interdomain traffic engineering can lead to the routing instability. We prove that, if there is no *provider-customer loop* in the network, each AS follows the static *typical* export policy, and AS ranking of routes follows the *standard joint-route preference policy*, then the convergence and uniqueness of route selection for egress interdomain traffic engineering can be guaranteed. This result is particularly pleasant and somehow surprising in that the conditions of the result are highly likely to be satisfied in the current Internet due to the ISP economy of the current Internet.

We complement the preceding analysis with extensive simulations to investigate the likelihood of instability when the three conditions are violated (*e.g.*, when some ASes give non-economic considerations higher priority over economic considerations). Specifically, we use current Internet BGP routing tables to infer the AS-level topology and AS business relationships. We then conduct simulations using the inferred Internet topology. We show that even with a small number of ASes coordinating route selection for just a small number of destinations, we can observe instability.

Although the preceding stability results are surprisingly pleasant and elegant, practice poses further challenges in analyzing interdomain routing stability. First, the previous studies focus on a specific interdomain route selection algorithm (*e.g.*, the BGP-based greedy route selection algorithm such as SPVP [29]). As a result, factors such as route dampening, which are present in routing practice, are not easily allowed in previous analysis. Although conceptually such factors might not change the conclusions of previous analysis, an analytical framework is still missing. Second, the previous studies focus on local policies which rank only the egress routes; that is, they assume that the local ranking of egress routes at each autonomous system is independent of the inbound traffic pattern of the AS. This independence is justified when the inbound traffic of an AS

is relatively constant. However, in practice, the local policies of ASes may involve both the egress routes and the pattern of inbound traffic, introducing unexpected interaction.

Specifically, an AS may rank egress routes depending on the pattern of inbound traffic. If this happens, we say that the local policy of the AS depends on the inbound traffic pattern, or inbound traffic for short. We also say that the local policy of the AS is inbound-traffic-dependent, or inbound-dependent for short. One way such inbound-dependent route selection can happen is that the operator of the AS observes traffic demand, and manually reconfigures the local preference values of the two routes. Such inbound-dependent route selection can also be implemented automatically, with a traffic engineering algorithm based on an estimated traffic demand matrix. In the last few years, several traffic-demand-matrix-based traffic engineering algorithms have been proposed (*e.g.*, [5, 26]). Although such algorithms have been shown to be effective, the evaluations often assume that the inbound traffic is constant (*e.g.*, the route selection of the AS does not change the inbound traffic). Furthermore, an AS may not only passively react to given inbound traffic, but also actively try to influence the pattern of the inbound traffic (*e.g.*, attracting more customer traffic, and/or load-balancing inbound traffic). The large number of prepended prefixes in BGP routing tables [6] indicates that it is a common practice that ASes try to influence inbound traffic. Recently, we have conducted an email survey of ISPs, and the results indicate that ISPs not only passively react to inbound traffic, but also actively try to influence the pattern of inbound traffic.

In this paper, we further analyze the stability of interdomain routing under the general model that the local preference of an AS depends on not only its egress routes to the destinations but also its inbound traffic pattern. Furthermore, instead of studying a specific route selection algorithm, we study a large class of route selection algorithms which are characterized by their asymptotic behaviors.

Specifically, we show that the common route selection algorithms of choosing the best routes according to the traffic demand matrix of the preceding period could lead to instability, when the route selection of an AS can change its inbound traffic pattern. This instability happens even when all constraints on interdomain routing imposed by business considerations [22] are satisfied, and just a single AS is using such an algorithm. We say that such instability is caused by traffic-route mis-association, and it is an example of instability caused by route selection algorithms. As a remedy, an AS should adopt a route selection algorithm which estimates inbound traffic in such a way that the estimated inbound traffic is truly the result of the chosen egress route.

We then analyze the stability of a network where ASes run any reasonable route selection algorithms which we call *rational route selection algorithms*. The definition of a rational route selection algorithm depends only on the asymptotic behavior of the algorithm. There are several advantages in conducting stability analysis based on the general notion of rational route selection algorithms. First, it allows us to establish stronger positive results in two senses: 1) it allows us to prove the stability of a heterogeneous network where different ASes can run different route selection algorithms, so long all of the algorithms are rational; 2) since the notion of a rational route selection algorithm is defined by its asymptotic behavior, if variations to a route selection algorithm do not change its asymptotic behavior (*e.g.*, non-persistent route dampening), the route selection algorithm is still rational, and thus the stability result still holds. Second, it allows us to establish stronger negative results; for example, if we show that a network is unstable under *any*

rational route selection algorithms, it is stronger than to show that a network is unstable under a specific route selection algorithm.

In particular, we derive a sufficient condition to guarantee routing convergence under the general model that the local preference of an AS depends on not only its egress routes to the destinations but also its inbound traffic pattern. This condition applies to any network so long the route selection algorithms of the ASes are rational route selection algorithms. The condition also allows us to predict potential routes. We also show that there exist networks which can have persistent route oscillations even when the local policy of each AS follows the constraints imposed by business considerations, and can adopt *any* one of the rational route selection algorithms. This result clearly demonstrates the intrinsic challenges of route selection for interdomain routing.

The rest of this paper is organized as follows. In Section 2, we discuss related work. In Section 3, we study route selection for egress interdomain traffic engineering. In Section 4, we show that the constraints imposed by Internet business considerations lead to unique stable egress route selection for interdomain traffic engineering. In Section 5, we present evaluations of route selection for egress interdomain traffic engineering. In Section 6, we study general route selection for interdomain traffic engineering. Our conclusion and future work are in Section 7.

## 2 Related Work

There is a large body of literature on interdomain route selection where each destination is considered separately. In particular, researchers have conducted extensive evaluations (*e.g.*, [16, 27, 34, 35, 53]) and theoretical analysis (*e.g.*, [11, 28, 31, 32, 47]) on the stability of BGP route selection. In particular, Griffin, Shepherd, and Wilfong [29] show that “policy disputes” can cause persistent route oscillations. Griffin and Wilfong [30] then propose a protocol called SPVP3 that can detect oscillations caused by policy disputes at run time using “path history.” SPVP3 is guaranteed to converge if routes whose path history contain cycles are suppressed. Feamster and Johari and Balakrishnan [11] study routing systems with ranking independence and unrestricted filtering; they use “dispute ring,” a specialized dispute wheel, to show that any routing system that has a dispute ring is not safe under filtering and that ASes are essentially required to rank routes based on AS-path lengths in order to guarantee convergence. Gao and Rexford [22, 24] observe that, if every AS considers each of its neighbors as either a customer, a provider, or a peer, and obeys certain local constraints on preference and export policies, then BGP is guaranteed to converge. Generalizing the above commercial relationships of ISPs to a class-based system, Jaggard and Ramachandran [31] show that a global constraint that guarantees convergence can be enforced by a distributed algorithm. The major difference between our model and the previous studies is that the previous studies consider only a network where there is no AS whose routing policies require it to coordinate the route selection of multiple destinations. Thus the route for each destination can be chosen regardless of the chosen/available routes of other destinations. As a result, the routing decisions for the destinations can be separated. In this paper, we investigate the effects of the coordination of route selection among multiple destinations, which is an essential feature of interdomain traffic engineering that has been missing in previous studies.

Traffic engineering has traditionally been focused on intra-domain (for a good survey, please

see [17, 18]). There is an increasing interest in tuning BGP attributes for interdomain traffic engineering [45]. However, most of the previous work focuses on the configuration of either a single AS (*e.g.*, [5, 9, 26]) or between two neighboring ASes. In particular, researchers have conducted extensive theoretical analysis (*e.g.*, [33]) and experimental evaluations (*e.g.*, [50, 51]) of hot-potato routing, which is a scheme of exit route selection between two ASes. Recognizing the potential unpredictable nature of interdomain BGP traffic engineering involving multiple ASes, Feamster *et al.* [8] propose guidelines to restrict route selection so that its impact on the traffic flow is predictable.

There is another line of research that proposes extensions/alternatives to BGP (*e.g.*, the mechanism-design approach by Feigenbaum *et al.* [12–14], the negotiation protocol by Mahajan *et al.* [38–40], the BGP pricing approach by Afergan and Wroclawski [1], the Hybrid Link-state Path-vector (HLP) approach by Subramanian *et al.* [49]). To assess the applicability and effectiveness of these new solutions to interdomain traffic engineering, we need to understand the intrinsic problems of route selection for interdomain traffic engineering. The objective of this paper is to pinpoint these problems; thus it can serve as a motivation for the initiation of these studies. It could also provide new insight to these studies. For example, we will show that there may not be Pareto optimal solutions if negotiation happens only between two neighboring ASes; this indicates that, for efficient route selection, current proposals of negotiation protocols (*e.g.*, [38]) need to be extended to handle much more general settings.

The interaction of interdomain routing and inbound traffic starts to receive some attention lately [25, 56]. However, the focus of previous studies is on prepending. In [56], Wang *et al.* characterize the stability of inbound-dependent route selection. However, their study focuses on prepending and their specific algorithm. Unlike [56], we focus on route selection, since we feel that the effects of prepending cannot be guaranteed since an AS can choose to ignore the effects of prepending. Also, we investigate the existence and nonexistence of stable route selection for general algorithms, instead of a specific algorithm. To model potential AS behaviors, we adopt a general, rational, learning model. This model is motivated by general game-theoretical, rational algorithms (*e.g.*, adaptive and sophisticated learning algorithms [42]). In particular, our model is inspired by the adaptive learning model of Milgrom and Roberts [42], and the reasonable learning model of Friedman and Shenker [19–21].

## 3 Route Selection for Egress Interdomain Traffic Engineering

### 3.1 Motivation

As we pointed out in Section 1, major ISPs are already coordinating the route selection of multiple destinations in their interdomain route selection. A very simple illustrative example is shown in Figure 1.

In this example, the majority of the traffic of AS  $S$  goes to two destinations  $D_1$  and  $D_2$ . Assume  $S$  wants to balance its outgoing traffic. Thus, it wants to choose a combination of routes for destinations  $D_1$  and  $D_2$  such that they use different neighbors, if possible, to have low utilization on the two links  $SA$  and  $SB$ . We refer to a combination of routes for  $D_1$  and  $D_2$  as a *route*

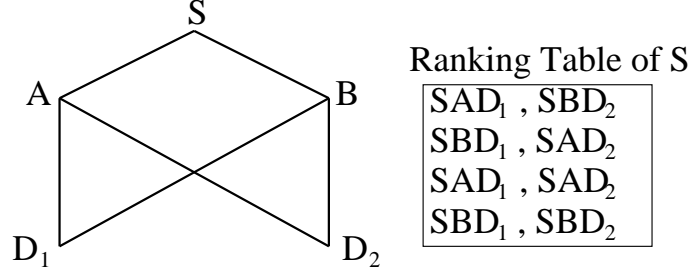


Figure 1: Egress load balancing: an example motivating the need for destination path interaction.

*profile*. Since  $S$  may not know in advance the routes it will learn from its neighbors  $A$  and  $B$ , or the routes that  $A$  and  $B$  will export to  $S$  can be dynamic given network dynamics,  $S$  needs an automatic method to pick the best route profile, according to currently available routes. One way  $S$  specifies its preference is to define an interdomain traffic engineering objective function (*e.g.*, minimize the maximum of the utilization of the two links for this case). An advantage of using an objective function is its compact representation. Given the objective function, link capacities, and traffic demands, a traffic engineering program searches for the best route profile automatically and dynamically, according to currently available routes. The preference can also be specified by a policy language. An example policy can be: if  $D_1$  and  $D_2$  use different links, assign a base local preference of 100; otherwise, a base local preference of 0. If  $D_1$  uses link  $SA$ , add 10 to local preference. If  $D_2$  uses link  $SB$ , add 5 to local preference. The program picks the available route profile with the highest local preference. For generality, we assume a ranking table at each AS, which lists, in decreasing order, all of the potential route profiles. An example route ranking table for  $S$  is shown in Figure 1, where each row is a route profile, *i.e.*, a combination of routes for  $D_1$  and  $D_2$ . For example, the best route profile for  $S$  is  $(SAD_1, SAD_2)$ ; *i.e.*,  $S$  uses  $SAD_1$  for destination  $D_1$ , and  $SAD_2$  for destination  $D_2$ . The worst route profile is  $SBD_1$  and  $SBD_2$ . Thus, if the route profile  $(SAD_1, SAD_2)$  is available,  $S$  will choose it. On the other hand, if the only available route profile is  $(SBD_1, SBD_2)$ ,  $S$  has no choice but to use it.

### 3.2 Problem Formulation

We first state the assumptions we made in this section. We assume a connected network with a set  $\mathcal{S}$  of source ASes and a set  $\mathcal{D}$  of destination ASes. We assume that the underlying network infrastructure is stable so that we can focus on the effects of interdomain traffic engineering policies. We assume that there is only one link between two neighboring ASes; that is, we consider eBGP and assume a consistent iBGP. Each AS chooses the best available routes in order to achieve its own interdomain traffic engineering objectives. For scalability, an AS may coordinate the route selection of only a subset of its destinations (*e.g.*, the “elephants” [15,44,52]), instead of all of the destinations. Our presentation assumes that the route selection of all destinations is coordinated; the scenarios that the route selection of some of the destinations is independent of other destinations are just special cases. We assume that each AS has a static export policy (*e.g.*, dictated by business contracts or common practice). In this section, we assume that, the preference of an AS

depends only on the route from the AS itself to the destinations. In other words, the ASes are conducting *egress interdomain traffic engineering*, which is one of the major tasks of ISP interdomain traffic engineering [10]. In Section 6 we will further extend this model and study route selection for general interdomain traffic engineering, in which case the route from each source to the AS itself also matters. Note also that in a more general case, the preference of an AS on a route may also depend on routes that do not pass through the AS itself. For example, these routes may share common links with the route chosen by this AS and thus cause congestion. We do not consider this problem and leave it to the study of the general congestion game [7].

Now we formally define the stable route selection for egress interdomain traffic engineering problem. For a list of notations, please see Appendix A.

The network topology is represented by a simple, undirected graph  $G = (V, E)$ , where  $V = \{1, \dots, N\}$  is the set of ASes and  $E$  is the set of interdomain links.

A path in  $G$  is either the empty path, denoted by  $\epsilon$ , or a sequence of ASes  $(v_k, v_{k-1}, \dots, v_1, v_0)$ , where  $k \geq 0$  is the length of the path, such that  $(v_i, v_{i-1}) \in E$  for  $i = k, k-1, \dots, 1$ . Note that if  $k = 0$ , then  $(v_0)$  represents the trivial path from  $v_0$  to itself. Each nonempty path  $P = (v_k, v_{k-1}, \dots, v_1, v_0)$  has a direction from  $v_k$  to  $v_0$ . If  $P$  and  $Q$  are two nonempty paths such that the first AS in  $Q$  is the same as the last AS in  $P$ , then  $PQ$  denotes the path formed by the *concatenation* of these two paths. We extend this with the convention that  $\epsilon P = P\epsilon = P$  for any path  $P$ . If  $P = (v_k, v_{k-1}, \dots, v_1, v_0)$  is a nonempty path, then for  $k \geq i > j \geq 1$ ,  $P[v_i, v_j]$  denotes the subpath of  $P$  from  $v_i$  to  $v_j$ .

We denote by  $R$  the set of all paths in  $G$ . For each  $i \in V$ , we denote by  $R_{i \rightarrow}$  the set of paths originating from  $i$ , and by  $R_{\rightarrow i}$  the set of paths terminating at  $i$ . Also, for any  $i, j \in V$ ,  $R_{i \rightarrow j} = R_{i \rightarrow} \cap R_{\rightarrow j}$  denotes the set of paths from  $i$  to  $j$ . This notation also extends to a set of sources and a set of destinations. If  $\mathcal{S} \subseteq V$  is a set of sources and  $\mathcal{D} \subseteq V$  is a set of destinations, then  $R_{\mathcal{S} \rightarrow \mathcal{D}}$  denotes the set of paths from any AS in  $\mathcal{S}$  to any AS in  $\mathcal{D}$ . In addition, if  $\mathcal{P}$  is a set of paths, then we denote by  $\mathcal{P}_{\mathcal{S} \rightarrow \mathcal{D}} = \mathcal{P} \cap R_{\mathcal{S} \rightarrow \mathcal{D}}$  the subset of paths of  $\mathcal{P}$  from any AS in  $\mathcal{S}$  to any AS in  $\mathcal{D}$ .

Suppose  $i$  and  $j$  are two neighboring ASes. As a path  $P$  is exported from  $j$  and imported into  $i$ , it undergoes two transformations. First,  $P_1 = \text{export}(i, j, P)$  represents the application of export policies of  $j$  to  $P$ , which includes possibly prepending  $j$  multiple times to  $P$  or filtering out  $P$  altogether ( $P_1 = \epsilon$ ). Second,  $P_2 = \text{import}(i, j, P_1)$  represents the application of import policies of  $i$  to  $P_1$ . In particular, import policies at  $i$  will filter out any path that contains  $i$  itself ( $P_2 = \epsilon$ ). The collective effects of these transformations can be represented by the *peering transformation*,  $\text{pt}(i, j, P)$ , defined as

$$\text{pt}(i, j, P) = \begin{cases} \text{import}(i, j, \text{export}(i, j, P)) & \text{if } (i, j) \in E, \\ \epsilon & \text{otherwise.} \end{cases}$$

The peering transformation represents the import/export policies of all ASes in the network. Note that in the above definition, we extend the domain of  $\text{pt}$  to all pairs of ASes by setting  $\text{pt}(i, j, P) = \epsilon$  if  $i$  and  $j$  are not neighbors.

Each AS  $i \in V$  has a set  $\mathcal{D}_i \subseteq V$  of destinations, and attempts to establish a path to each destination  $j \in \mathcal{D}_i$ . A *network route selection* is a function  $r$  that maps each pair of ASes  $i \in V$

and  $j \in \mathcal{D}_i$  to a path  $r(i, j) \in R_{i \rightarrow j}$ . We interpret  $r(i, j) = \epsilon$  to mean that  $i$  is not assigned a path to  $j$ . We denote by  $\mathcal{R}$  the set of all possible network route selections. When we restrict our attention to the route selection of AS  $i$  alone, we shall refer to the restriction of  $r$  on  $i$  and  $\mathcal{D}_i$  as the *route profile* for AS  $i$ , denoted by  $r_i$ . In addition, for any  $\mathcal{D} \subseteq \mathcal{D}_i$ , we refer to the further restriction of  $r_i$  on  $\mathcal{D}$  as the *partial route profile* from AS  $i$  to destinations in  $\mathcal{D}$ , denoted by  $r_i^{\mathcal{D}}$ . We denote by  $\mathcal{R}_i$  the set of all possible route profiles for AS  $i$ , and by  $\mathcal{R}_i^{\mathcal{D}}$  the set of all possible partial route profiles from AS  $i$  to destinations in  $\mathcal{D}$ . Furthermore, for a set  $\mathcal{P}$  of paths, We denote by  $\mathcal{R}_i^{\mathcal{D}}(\mathcal{P})$  the set of all possible partial route profiles for AS  $i$  with paths from  $i$  to destinations in  $\mathcal{D}$  drawn from  $\mathcal{P}$ ; that is

$$\mathcal{R}_i^{\mathcal{D}}(\mathcal{P}) = \{r_i^{\mathcal{D}} | r_i^{\mathcal{D}}(j) \in \mathcal{P}_{i \rightarrow j}, \forall j \in \mathcal{D}\}.$$

Note that in the above definition, we do *not* require the routes in a network route selection to be consistent; that is, if  $r_i(k) = (i, j)P$ , it is not necessary that  $r_j(k) = P$ .

The above definitions lead to useful equivalent representations of network route selections and route profiles. First, a network route selection  $r$  can be represented as  $r = (r_i, r_{-i})$ , where  $r_{-i} = (r_j)_{j \neq i}$  denotes the *combined route profiles* of all ASes except  $i$ . The route profile of AS  $j \neq i$  in  $r_{-i}$  is denoted by  $(r_{-i})_j$ . We denote by  $\mathcal{R}_{-i}$  the set of all possible combined route profiles of all ASes except  $i$ ; that is,  $\mathcal{R}_{-i} = \{r_{-i} | (r_{-i})_j \in \mathcal{R}_j, \forall j \neq i\}$ . Second, network route selections and (combined) route profiles can be treated as sets of paths. Specifically, a network route selection  $r$ , a route profile  $r_i$  and a combined route profile  $r_{-i}$  are equivalent to the sets of paths  $\{r(i, j) | i \in V, j \in \mathcal{D}_i\}$ ,  $\{r_i(j) | j \in \mathcal{D}_i\}$ , and  $\{(r_{-i})_j(k) | k \in \mathcal{D}_j, j \neq i\}$ , respectively. This equivalent representation is particularly convenient in some operators defined on sets of paths. For example, we can simply use  $r_{-i}$  as an argument to such an operator, where actually the argument is  $\{(r_{-i})_j(k) | k \in \mathcal{D}_j, j \neq i\}$ .

For the purpose of traffic engineering, AS  $i$  would like to coordinate its routing for destinations in  $\mathcal{D}_i$ . In general, however, it is unlikely and impractical for AS  $i$  to coordinate its routing for all destinations in  $\mathcal{D}_i$  as a whole. A more general and reasonable approach for AS  $i$  is to partition the destinations in  $\mathcal{D}_i$  into a family of disjoint subsets  $\mathcal{D}_{ik}$ , for  $k = 1, \dots, N_i$ . For each subset  $\mathcal{D}_{ik}$ , AS  $i$  chooses routes jointly for all destinations in  $\mathcal{D}_{ik}$ . This coordinated routing for destinations in  $\mathcal{D}_{ik}$  can be captured by a *route selection function*  $\sigma_i^{\mathcal{D}_{ik}}$ , which maps a set of available paths to a partial route profile from  $i$  to  $\mathcal{D}_{ik}$ . Given a set  $\mathcal{P}$  of available paths, AS  $i$ 's chosen routes to destinations in  $\mathcal{D}_{ik}$  are given by a partial route profile

$$r_i^{\mathcal{D}_{ik}} = \sigma_i^{\mathcal{D}_{ik}}(\mathcal{P}), \text{ such that } r_i^{\mathcal{D}_{ik}}(j) \in \mathcal{P}_{i \rightarrow j}, \forall j \in \mathcal{D}_{ik}.$$

In this paper, we focus on the model of route selection which can be represented by a linear preference order. Specifically, each AS  $i$  has a ranking function  $\lambda_i^{\mathcal{D}_{ik}}$  for each  $\mathcal{D}_{ik}$ , which maps any possible partial route profile from  $i$  to  $\mathcal{D}_{ik}$  to a totally ordered set  $\Lambda$ . Given a set  $\mathcal{P}$  of available paths, the route selection function  $\sigma_i^{\mathcal{D}_{ik}}$  simply selects the highest ranked possible partial route profile, *i.e.*

$$\sigma_i^{\mathcal{D}_{ik}}(\mathcal{P}) = \arg \max_{r \in \mathcal{R}_i^{\mathcal{D}_{ik}}(\mathcal{P})} \lambda_i^{\mathcal{D}_{ik}}(r).$$

The route profile  $r_i$  for AS  $i$  is determined by selecting partial route profiles for all  $\mathcal{D}_{ik}$ 's *independently*. The overall route selection behavior of AS  $i$  is represented by a route selection function  $\sigma_i$

defined as

$$r_i = \sigma_i(\mathcal{P}), \text{ such that } r_i^{\mathcal{D}_{ik}} = \sigma_i^{\mathcal{D}_{ik}}(\mathcal{P}), \forall k = 1, \dots, N_i.$$

We emphasize again that the ranking functions  $\lambda_i^{\mathcal{D}_{ik}}$  are just general representations of some more compact representations such as objective functions or policy languages.

When the subset  $\mathcal{D}_{ik}$  is clear from context, we abbreviate  $r_i^{\mathcal{D}_{ik}}$  as  $r_i^k$ ,  $\sigma_i^{\mathcal{D}_{ik}}$  as  $\sigma_i^k$ , and  $\lambda_i^{\mathcal{D}_{ik}}$  as  $\lambda_i^k$ .

A BGP system is a quintuple  $S = (G, \text{pt}, \sigma, \mathbb{D}, \tilde{\mathbb{P}})$ , where  $G = (V, E)$  is the topology of a network,  $\text{pt}$  is a peering transformation defined on  $G$ ,  $\sigma_i$  is the route selection function of AS  $i$ , and  $\tilde{\mathbb{P}}_i$  is the set of *feasible* paths from  $i$  to destinations in  $\mathcal{D}_i$ .

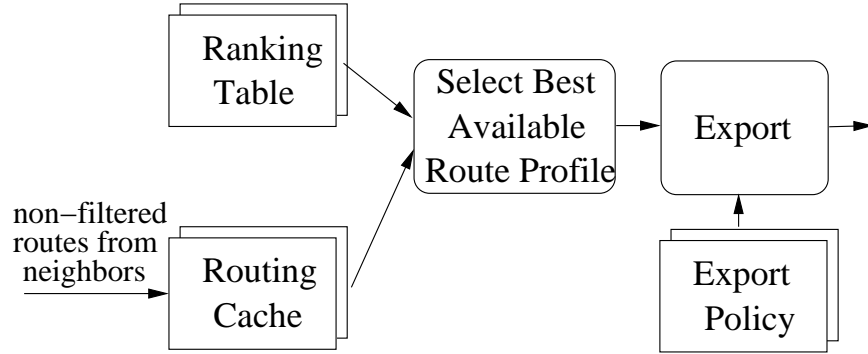


Figure 2: The protocol/process model of route selection for interdomain traffic engineering.

Figure 2 shows the standard protocol/process model of interdomain route selection [22, 28, 29, 32, 47], naturally extended to multiple destinations. Specifically, each AS maintains a routing cache  $A_i$  of currently available routes exported by its neighbors. AS  $i$  selects a route profile  $r_i$  from its routing cache  $A_i$  using its route selection function  $\sigma_i$  as defined above<sup>1</sup>, which will then be used by  $i$  to route packets. Sometime we refer to this chosen route profile as the *installed* route profile. If  $r_i(j)$  is different from the previously selected route to  $j$ ,  $i$  then withdraws the previous route, and exports the new route to the neighbors that are allowed to receive this route according to  $i$ 's export policy. We assume that BGP route update messages between neighboring ASes are delivered in FIFO order and reliably. This is reasonable as the messages are sent via TCP. We also assume that each message will be processed in a bounded time.

Given the above description of the protocol/process model of interdomain route selection, we now define the notion of a *stable* network route selection. For a given network route selection  $r$ , the set  $\text{candidates}(i, r)$  consists of all available paths at AS  $i$  that can be formed by extending the routes chosen by neighbors of  $i$ ; that is,

$$\text{candidates}(i, r) = \{\text{pt}(i, j, r_j(k)) \mid (i, j) \in E \text{ and } k \in \mathcal{D}_j\}.$$

The network route selection  $r$  is *stable* if no AS  $i$  can choose a higher ranked route profile from

<sup>1</sup>Due to computational complexity, for some formulations of interdomain traffic engineering, it could be the case that only approximate solutions can be obtained. We leave this consideration as future work.

candidates( $i, r$ ); formally,  $r$  is stable if and only if

$$r_i = \sigma_i(\text{candidates}(i, r)), \text{ for all } i \in V.$$

We also call a stable network route selection a *stable route solution* or *solution* for short.

Finally, a network is *robust* if BGP protocol is guaranteed to converge even with arbitrary node/link failures.

### 3.3 Multi-Destination Interactions Can Cause Instability

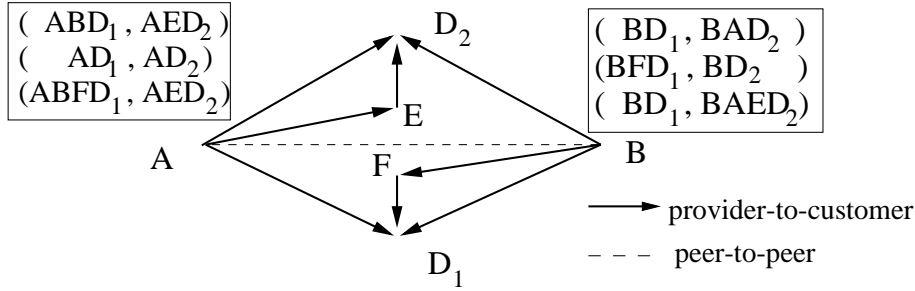


Figure 3: An example network which has no stable route selection.

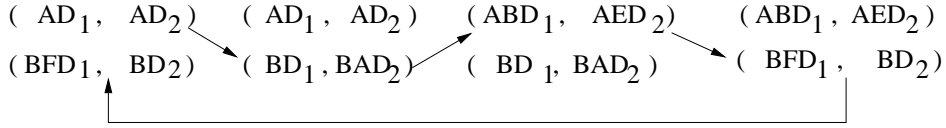


Figure 4: The BGP update process of the network in Figure 3.

As we pointed out in Section 1, the interaction of the routing of multiple destinations due to interdomain traffic engineering can cause routing instability. The network shown in Figure 3 is one such interesting example. For clarity, we show only the highest three route profiles of  $A$  and  $B$ . The export policies of  $A$  and  $B$  follow the *typical export policies* [22, 23]: 1) each AS exports to its providers its own routes and those it learned from its customers, but does not export to its providers the routes it learned from its peers or other providers; 2) each AS exports to its customers its own routes and any routes it learned from others; 3) each AS exports to its peers its own routes and those it learned from its customers, but does not export those it learned from its providers or other peers.

We first consider each destination separately. For destination  $D_1$ , the two routes for  $D_1$  contained in the two highest route profiles of  $A$  are  $ABD_1$  and  $AD_1$ ; the two routes for  $D_1$  contained in the two highest route profiles of  $B$  are  $BD_1$  and  $BFD_1$ . Consider this combination of route preference for  $D_1$ . The network has the stable route solution of  $ABD_1$  and  $BD_1$  for  $A$  and  $B$ , respectively. One can also verify that if we consider  $D_2$  alone, the network has the stable route solution of  $AED_2$  and  $BD_2$  for  $A$  and  $B$ , respectively. Thus, if there were no interaction among

destinations,  $A$  and  $B$  would settle to the stable solutions of  $(ABD_1, AED_2)$  and  $(BD_1, BD_2)$ , respectively.

Next we consider destination interaction. The above solutions obtained by considering each destination alone are no longer stable. For example,  $B$  will not choose  $(BD_1, BD_2)$  since this route profile has a low rank. One can verify that the network has no stable solution at all. Specifically, we observe that the export policies of the ASes make the route profile  $(AD_1, AD_2)$  always available to  $A$ . Thus to see that the network has no stable solutions, we just need to verify that there is no stable route solution when  $A$  chooses  $(AD_1, AD_2)$  or  $(ABD_1, AED_2)$ . Clearly, there is no stable solution for  $(AD_1, AD_2)$  since if  $A$  chooses  $(AD_1, AD_2)$ ,  $B$  will choose  $(BD_1, BAD_2)$ ; this causes  $A$  to change to  $(ABD_1, AED_2)$ . However, there will be no stable route selection for  $(ABD_1, AED_2)$  neither. To make  $(ABD_1, AED_2)$  available to  $A$ ,  $B$  must choose  $BD_1$  for  $D_1$ . Since  $(BD_1, BAD_2)$  is always available to  $B$ , it must be the case that  $B$  chooses  $(BD_1, BAD_2)$ . However, this requires  $A$  to choose  $AD_2$ , which is inconsistent with  $(ABD_1, AED_2)$ . Thus, the network has no stable route selections due to destination interaction! Figure 4 shows the BGP update process.

### 3.4 Stable, Robust Route Selection and Protocol Convergence

Given that multi-destination interaction can result in no stable route selection, in this section, we derive a sufficient condition that can guarantee stable, robust route selection and protocol convergence.

#### 3.4.1 Representation of Protocol Execution

Based on the protocol/process model described in subsection 3.2, we adopt the following representation of an arbitrary protocol execution. We assume that the BGP update messages are delivered reliably and in FIFO order, and the protocol is fair [29]. We assume a total ordering of events; that is, we assign a unique index from  $T = \{0, 1, 2, \dots\}$  to each event so that the assignment is consistent with the logical “happen before” relation among events [37]. We have the following three types of events in our system: type 1) send a route update message; type 2) receive a route update message and update the route in the cache that is affected by the route update message; and type 3) select the highest-ranked route profile and install it as the current route profile. For ease of description, we refer to the ordering as *time* from now on. Specifically, when we write time  $t$ , we mean the index  $t$  assigned to an event in the total ordering. Let  $r[t]$  be the network route selection at time  $t$ , then an arbitrary execution of the protocol can be represented by a sequence of network route selections,  $\{r[t]\}_{t \in T}$ .

#### 3.4.2 Self-contained BGP Subsystem

A stable network route selection as defined in subsection 3.2 is a *network-wide* concept, where the route from any source to any destination is required to be stable. In a large network, however, it may well be the case that some routes have become stable, while others are still oscillating. It is of theoretical and practical interests, therefore, to consider *partial convergence* in a large network.

To capture this intuitive idea of partial convergence, we introduce the notion of a *self-contained BGP subsystem*. A BGP subsystem  $S = (G, \text{pt}, \sigma, D, \tilde{P})$  is a BGP system where the set  $D_i$  may not contain all of the destinations that AS  $i$  attempts to establish a route to. In a BGP subsystem, we restrict our attention to a subset of destinations for AS  $i$ , particularly those to which the routes may become stable. AS  $i$  may have routes to other destinations, but these routes are not of our interests. We do have a requirement, however, on which destinations and routes can be left out. Intuitively, we wish to omit only those routes that will not be chosen after some finite time. Formally, the BGP subsystem  $S$  is *self-contained* if there exists  $P_i \subseteq \tilde{P}_i$  for all  $i \in V$ , such that

1. there exists  $t$ , such that for all  $t' > t$  and  $i \in V$ ,  $r_i[t'] \in R(i, D_i, P_i)$ ;
2.  $P_i \subseteq \{\text{pt}(i, j, Q) \mid (i, j) \in E, Q \in P_j\}$ , for all  $i \in V$ .

A self-contained BGP subsystem is represented by  $S = (G, \text{pt}, \sigma, D, \tilde{P}, P)$ , or sometimes  $S_P$  for short when the underlying BGP system  $S$  is clear from context.

### 3.4.3 P-graph and P-cycle

We now introduce the notion of a P-graph to capture the interaction of interdomain traffic engineering policies of multiple ASes in a self-contained BGP subsystem  $S = (G, \text{pt}, \sigma, D, \tilde{P}, P)$ . The notion of a P-graph is motivated by the partial order graph of Griffin *et al.* [28], but generalized to interdomain traffic engineering.

A P-graph is a directed graph constructed as follows. For each AS  $i$  and each  $D_{ik}$ , there is a node which corresponds to each possible partial route profile  $r_i^{D_{ik}} \in R(i, D_{ik}, P_i)$ . Note that we do not consider partial profile formed by paths in  $\tilde{P}_i \setminus P_i$ . There are two types of directed edges in a P-graph. The first type of edges are *improvement edges*. There is an improvement edge from node  $\tilde{r}_i^{D_{ik}}$  to  $\hat{r}_i^{D_{ik}}$  if  $i$  prefers  $\hat{r}_i^{D_{ik}}$  to  $\tilde{r}_i^{D_{ik}}$  ( $\lambda_i^{D_{ik}}(\hat{r}_i^{D_{ik}}) > \lambda_i^{D_{ik}}(\tilde{r}_i^{D_{ik}})$ ). The second type of edges are *sub-path edges*. There is a destination  $d$  sub-path edge from a node  $r_i^{D_{ik}}$  to another node  $r_j^{D_{jl}}$  if the path  $r_j^{D_{jl}}(d)$  from  $j$  to  $d$  is a sub path of the path  $r_i^{D_{ik}}(d)$  from  $i$  to  $d$ . Note that in this case  $d \in D_{ik} \cap D_{jl}$ .

A P-cycle is a loop in the P-graph of the following special format: one or more improvement edges, followed by one or more sub-path edges of the same destination, then followed by one or more improvement edges, and so on. For example, Figure 5 shows the P-graph and the P-cycle for the example of Figure 3. Note that there may be trivial loops in a P-graph which are not of the format of a P-cycle. For example, the loop consisting of  $(BD_1, BAD_2)$ ,  $(AD_1, AD_2)$  and  $(ABD_1, AED_2)$  is not a P-cycle, since there are two consecutive sub-path edges of different destinations.

### 3.4.4 BGP protocol convergence

We next apply the notion of P-graph to prove that BGP protocol converges. In doing so, we first prove the following lemma:

**Lemma 1** *If a self-contained BGP subsystem  $S_P$  does not converge, then there is a P-cycle in the corresponding P-graph.*

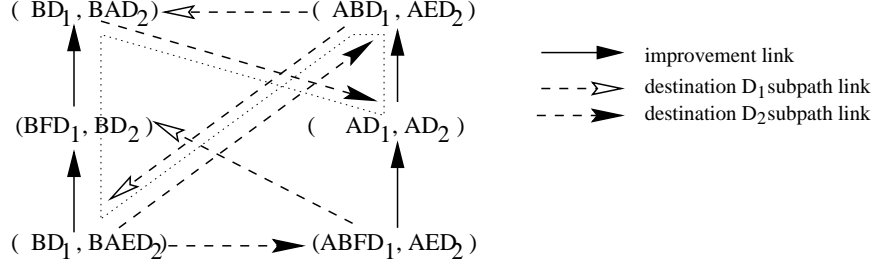


Figure 5: The P-graph and P-cycle of the network in Figure 3. For clarity, only a subset of route profiles and improvement links are shown.

**Proof:** As outlined in subsection 3.4.1, we represent an arbitrary execution of the protocol by a sequence of network route selections,  $\{r[t]\}_{t \in T}$ . Let  $r_u[t]$  be the route profile of AS  $u$  at time  $t$ . To simplify notation, in the following proof, we will abbreviate  $r_u^{D_{uk}}$  as  $r_u^k$ , and  $\lambda_u^{D_{uk}}$  as  $\lambda_u^k$ . Let  $R_u^k[\infty]$  be the set of partial route profiles which  $u$  chooses infinitely often for  $D_{uk}$ ; that is,  $R_u^k[\infty] = \cap_t \cup_{t' \geq t} \{r_u^k[t']\}$ . There exists  $t_f$  such that for any  $u$  and any  $t > t_f$ ,  $r_u^k[t] \in R_u^k[\infty]$ . In other words, after  $t_f$ , routes which are chosen only a finite number of times will no longer appear. It follows from condition 1 of a self-contained BGP subsystem that  $R_u^k[\infty] \in R(u, D_{uk}, P_u)$ . If the BGP process does not converge, then there exists a set  $O$  of ASes such that for each AS  $u \in O$ ,  $|R_u^k[\infty]| \geq 2$  for some  $k$ . These are the ASes that have persistent oscillating partial route profiles. Since the set  $R_u^k[\infty]$  is finite, we have the following observation:

**Proposition 2** *For any  $t > t_f$ , there exists  $t' > t$ , such that  $\lambda_u^k(r_u^k[t' - 1]) > \lambda_u^k(r_u^k[t'])$ ; that is,  $u$  will change from a higher-ranked partial route profile for destinations in  $D_{uk}$  to a lower-ranked one infinitely often.*

We shall construct a P-cycle as follows. We start from an arbitrary  $u_0 \in O$ . By Proposition 2, there exists  $k_0$  and  $t_0 > t_f$ , such that  $\lambda_{u_0}^{k_0}(r_{u_0}^{k_0}[t_0]) < \lambda_{u_0}^{k_0}(r_{u_0}^{k_0}[t_0 - 1])$ . Thus there is an improvement edge from partial route profile  $r_{u_0}^{k_0}[t_0]$  to  $r_{u_0}^{k_0}[t_0 - 1]$ .

The only reason for  $u_0$  to change from a higher-ranked partial route profile  $r_{u_0}^{k_0}[t_0 - 1]$  to a lower-ranked partial route profile  $r_{u_0}^{k_0}[t_0]$  is that, some time before  $t_0$ , a route  $P$  to some destination  $d \in D_{u_0 k_0}$  in  $r_{u_0}^{k_0}[t_0 - 1]$  is withdrawn by a BGP update message from  $u_0$ 's neighbor  $v$ . Let  $P[v, d]$  denote the sub-path of  $P$  from  $v$  to  $d$ . Thus there exists some  $t_f < t_1 < t_0$  and  $k$  such that  $v$  processes a type 3 event at time  $t_1$  and changes from a partial route profile  $r_v^k[t_1 - 1]$  containing  $P[v, d]$  to a partial route profile  $r_v^k[t_1]$  which does not contain  $P[v, d]$ .

There are two possible reasons for this change of  $v$ :

1. AS  $v$  ranks  $r_v^k[t_1]$  higher than  $r_v^k[t_1 - 1]$ . In this case, let  $\hat{r}_v^k = r_v^k[t_1 - 1]$  and  $\tilde{r}_v^k = r_v^k[t_1]$ , we have  $\lambda_v^k(\tilde{r}_v^k) < \lambda_v^k(\hat{r}_v^k)$ .
2. AS  $v$  ranks  $r_v^k[t_1]$  lower than  $r_v^k[t_1 - 1]$ . There are two sub-cases to consider:
  - (a) At time  $t_1$ , path  $P[v, d]$  is still available to  $v$ . In this case, let  $\hat{r}_v^k = r_v^k[t_1]$ , and let  $\tilde{r}_v^k$  be the partial route profile formed by replacing the route to destination  $d$  in  $r_v^k[t_1]$  with

$P[v, d]$ . Because  $\tilde{r}_v^k$  is an available route profile to  $v$  at time  $t_1$ , but  $v$  chooses  $\hat{r}_v^k$  instead, thus we have  $\lambda_v^k(\tilde{r}_v^k) < \lambda_v^k(\hat{r}_v^k)$ .

- (b) At time  $t_1$ , path  $P[v, d]$  is no longer available to  $v$ . Let  $P[v, d] = (v, w)P[w, d]$ , thus  $v$  must have received a BGP update message withdrawing  $P[w, d]$  from  $w$ . In this case, we take  $w$  as  $v$ , and repeat the argument. Since there are only a finite number of ASes on  $P$ , eventually we will come across  $v'$  where  $P[v', d]$  is still available to  $v'$ , in which case, we end up with case (2a).

Therefore, we can always find an AS  $v$ , a destination  $d \in D_{u_0k_0} \cap D_{vk}$ , and two partial route profiles  $\tilde{r}_v^k$  and  $\hat{r}_v^k$ , such that  $\tilde{r}_v^k(d)$  is a sub-path of  $r_{u_0}^{k_0}[t_0 - 1](d)$ , and  $\lambda_v^k(\tilde{r}_v^k) < \lambda_v^k(\hat{r}_v^k)$ . Because the BGP subsystem is self-contained, the fact that  $r_{u_0}^{k_0}[t_0 - 1] \in R(u_0, D_{u_0k_0}, P_{u_0})$  implies that both  $\tilde{r}_v^k$  and  $\hat{r}_v^k$  must also be in  $R(v, D_{vk}, P_v)$ . Thus, there is a destination  $d$  sub-path edge from  $r_{u_0}^{k_0}[t_0 - 1]$  to  $\tilde{r}_v^k$ , followed by an improvement edge from  $\tilde{r}_v^k$  to  $\hat{r}_v^k$ . After time  $t_1$ ,  $v$  may go through zero or more higher-ranked partial route profiles (thus one or more improvement edges in the P-graph). By proposition 2, eventually we will have a time  $t_2 > t_1$  such that,  $\lambda_v^k(r_v^k[t_2 - 1]) > \lambda_v^k(r_v^k[t_2])$ . Denote this  $v$  by  $u_1$ . Repeating the above reasoning on  $u_1$ 's change at time  $t_2$ , we can construct a path with alternating improvement edges and sub-path edges in the P-graph. Since the P-graph is a finite graph, eventually we will form a P-cycle. ■

Lemma 1 immediately leads to the following sufficient condition for convergence in a self-contained BGP subsystem.

**Corollary 3** *If the P-graph of a self-contained BGP subsystem  $S_P$  has no P-cycle, then the BGP protocol converges on destinations in  $D_i$  for all AS  $i \in V$ . In addition, let  $r^*$  be the network route selection after convergence, then  $r_i^* \in R(i, D_i, P_i)$  for all  $i \in V$ . Furthermore, the BGP subsystem is guaranteed to be robust.*

The robustness result follows easily from the fact that node/link failures will not introduce new P-cycle in P-graph.

One can extend the proof in [29] to show that, the converged route selection is stable (by proving that the state are kept consistent during protocol execution in a multiple destination setting); that is, each AS's route profile is the highest ranked among all valid route profiles that can be constructed from the exported highest ranked route profile of each of its neighbors (subject to export policies).

### 3.4.5 Composition of Self-contained BGP Subsystems

In order to establish BGP protocol convergence for the whole network, we can directly apply Corollary 3 on the whole BGP system, since the whole BGP system is trivially a self-contained BGP subsystem. Sometimes, however, it may be more convenient to first establish BGP protocol convergence for two or more non-trivial self-contained BGP subsystems, and then compose these subsystems to obtain convergence for the whole system.

There are two methods to compose two self-contained BGP subsystem  $S_1 = (G, \text{pt}, \sigma, D^{(1)}, \tilde{P}^{(1)}, P^{(1)})$  and  $S_2 = (G, \text{pt}, \sigma, D^{(2)}, \tilde{P}^{(2)}, P^{(2)})$ .

The first type of composition is *parallel* composition. In this type of composition,  $S_1$  and  $S_2$  are disjoint in the sense that BGP protocol convergence on  $D^{(1)}$  and  $D^{(2)}$  are totally independent. Specifically, parallel composition requires that  $D_i^{(1)} \cap D_i^{(2)} = \emptyset$ , for all  $i \in V$ . Note that this also implies that  $\tilde{P}_i^{(1)} \cap \tilde{P}_i^{(2)} = \emptyset$  and  $P_i^{(1)} \cap P_i^{(2)} = \emptyset$ . If we manage to establish convergence of  $S_1$  and  $S_2$ , it follows immediately that BGP protocol also converges on  $D_i^{(1)} \cup D_i^{(2)}$  for all  $i \in V$ .

The second type of composition is *sequential* composition. In this type of composition, BGP protocol converges on  $D^{(1)}$  first, and for any converged partial route profile for  $D^{(1)}$ , routes to destinations in  $D^{(2)}$  will also converge. Sequential composition requires two conditions. First,  $D_i^{(1)} \subseteq D_i^{(2)}$  for all  $i \in V$ . To define the second condition, for any stable route selection  $\hat{r}^{(1)}$  for  $S_1$ , let  $\tilde{P}^{(2)}|_{r^{(1)}=\hat{r}^{(1)}}$  be the subset of  $\tilde{P}^{(2)}$  such that paths to destinations in  $D^{(1)}$  is given by  $\hat{r}^{(1)}$ ; that is,  $\tilde{P}^{(2)}|_{r^{(1)}=\hat{r}^{(1)}}$  is the restriction of  $\tilde{P}^{(2)}$  by  $\hat{r}^{(1)}$ . Also define  $P^{(2)}|_{r^{(1)}=\hat{r}^{(1)}}$  in a similar way. Let  $S_2|_{r^{(1)}=\hat{r}^{(1)}}$  be the BGP subsystem  $(G, \text{pt}, \sigma, D^{(2)}, \tilde{P}^{(2)}|_{r^{(1)}=\hat{r}^{(1)}}, P^{(2)}|_{r^{(1)}=\hat{r}^{(1)}})$ . The second condition requires that for any  $\hat{r}^{(1)}$ ,  $S_2|_{r^{(1)}=\hat{r}^{(1)}}$  is a self-contained BGP subsystem. If we manage to show that BGP protocol converges on  $S_1$  and  $S_2|_{r^{(1)}=\hat{r}^{(1)}}$  for any stable  $\hat{r}^{(1)}$ , we can be sure that BGP protocol will eventually converge on  $D_i^{(2)}$  for all  $i \in V$ .

We will see an example of sequential composition of two self-contained BGP subsystems in section 4.

### 3.5 Network with non-Pareto Optimal Solution

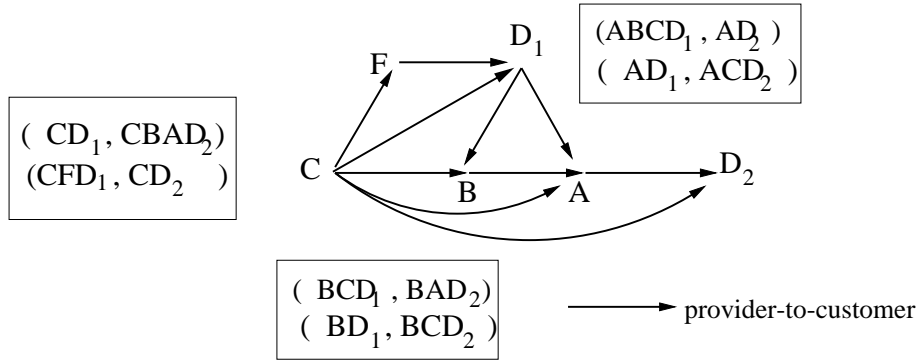


Figure 6: An example with two solutions but one of them is not Pareto optimal.

	A	B	C
Solution 1	$(ABCD_1, AD_2)$	$(BCD_1, BAD_2)$	$(CD_1, CBAD_2)$
Solution 2	$(AD_1, ACD_2)$	$(BD_1, BCD_2)$	$(CFD_1, CD_2)$

Figure 7: Two stable route selections for the network in Figure 6.

A network with stable solutions can have multiple solutions. The example in Figure 6 is one example.

This example is particularly interesting in that it has two stable route solutions, as shown in Figure 7, and the solution at the second row is not even Pareto optimal. Specifically, a stable route

solution is Pareto optimal if there does not exist another stable route solution where each AS has a higher ranked route profile. This example clearly demonstrates that to be effective, negotiation-based route selection [40] may involve more than two parties.

## 4 Stable Egress Route Selection without Global Coordination

The preceding section presents a sufficient condition to guarantee the convergence of route selection in a general network. The condition depends on checking P-cycle. In practice, it is difficult to obtain P-graph and check whether it contains a P-cycle. This is due to the fact that BGP is a distributed protocol, and generally ASes do not share their traffic engineering policies. Also, the preceding section considers general networks, while in the current Internet, the route selection policies of the ASes are not general, but are highly likely to be constrained by their business considerations. The question we will investigate in this section, therefore, is whether such constraints can lead to stability.

The constraints imposed by business considerations were first systematically studied by Gao and Rexford [22,24]. Specifically, they observed that the business considerations of ASes in current Internet imply that ASes follow the typical export policies (please see Section 3.3 for definition). Typical export policies imply that instead of arbitrary valid routes, valid routes in the Internet have the following patterns [22]: a provider-customer link can be followed only by provider-customer links, and a peer link can be followed only by provider-customer links. Accordingly, we divide the routes from an AS  $i$  to a destination  $d$  into three categories:

- *Customer route*: each link along a customer route is a provider-customer link.
- *Peer route*: the first link along a peer route is a peer link, and the remaining links are all provider-customer links.
- *Provider route*: the first link is a customer-provider link, and the remaining route consists of zero or multiple customer-provider links, followed by zero or one peer link, and then zero or multiple provider-customer links.

Hereafter, we denote by  $r_{i \rightarrow d}^C$ ,  $r_{i \rightarrow d}^E$ , and  $r_{i \rightarrow d}^P$  an instance of customer, peer, and provider route, respectively. Similarly, we denote the set of customer, peer, and provider routes by  $R_{i \rightarrow d}^C$ ,  $R_{i \rightarrow d}^E$ ,  $R_{i \rightarrow d}^P$ , respectively. We can further divide the set  $\mathcal{D}_i$  of destinations of an AS  $i$  into three categories, given that the above two constraints are satisfied:

- *Customer-reachable destinations*: these destinations are direct or transitive customers of AS  $i$ . Let  $\mathcal{D}_i^C$  be the set of customer-reachable destinations of AS  $i$ . We have  $\mathcal{D}_i^C = \{d | R_{i \rightarrow d}^C \neq \emptyset\}$ .
- *Peer-provider-reachable destinations*: these destinations are direct or transitive customers of one of AS  $i$ 's peers or providers, but they are not direct or transitive customers of AS  $i$ . Let  $\mathcal{D}_i^E = \{d | R_{i \rightarrow d}^E \neq \emptyset\} - \mathcal{D}_i^C$  be the set of peer-reachable destinations, and  $\mathcal{D}_i^P = \mathcal{D}_i - \mathcal{D}_i^C - \mathcal{D}_i^E$  be the set of provider-reachable destinations. We call  $\mathcal{D}_i^{EP} = \mathcal{D}_i - \mathcal{D}_i^C$  the set of peer-provider-reachable destinations of AS  $i$ .

Given the above definitions of different types of routes, Gao and Rexford [22, 24] observe that business considerations imply that an AS prefers customer routes over peer/provider routes. We call such route preference, namely, customer routes  $\succ$  peer/provider routes, *the standard individual-route preference policy*. Assuming the standard export policy, the standard individual-route preference policy, together with the assumption that there is no provider-customer loop (*PC-loop* for short) in the business relationships formed by ASes, Gao and Rexford prove that these conditions guarantee convergence in the global Internet.

A potential issue of their analysis is that their route selection model assumes that there is no coordination among destinations. However, as we discussed in the preceding sections, in the current Internet, ISPs are increasingly adopting coordinated route selection policies to achieve their interdomain traffic engineering objectives. Given such coordination, we need to re-evaluate AS route selection behaviors and investigate whether they lead to stability. Specifically, we need to reevaluate how the standard individual-route preference policy will change if an AS coordinates its routes to multiple destinations. If economics is the first consideration, then it is still reasonable that an AS will prefer customer routes over peer/provider routes, since customer routes bring in revenue. However, in the general case, now an AS may coordinate the route selection of multiple customer-reachable destinations. As for those peer-provider-reachable destinations, now an AS can jointly select routes for multiple such destinations to load balance, and to maintain peering traffic ratios.

Specifically, the route selection behavior of each AS  $i$  can be described by ranking functions  $\lambda_i^C$  and  $\lambda_i^{EP}$ . Note that we use  $C$  and  $EP$  instead of  $\mathcal{D}_i^C$  and  $\mathcal{D}_i^{EP}$  as superscripts to simplify notation, we will also abbreviate  $r_i^{\mathcal{D}_i^C}$  as  $r_i^C$ , and  $r_i^{\mathcal{D}_i^{EP}}$  as  $r_i^{EP}$ . Suppose  $\mathcal{A}_i$  is the set of paths available to  $i$ , then  $i$ 's selected route profile  $\hat{r}_i$  is given by

$$\hat{r}_i^C = \arg \max_{r_i^C \in \mathcal{R}_i^{\mathcal{D}_i^C}(\mathcal{A}_i)} \lambda_i^C(r_i^C), \quad (1)$$

$$\hat{r}_i^{EP} = \arg \max_{r_i^{EP} \in \mathcal{R}_i^{\mathcal{D}_i^{EP}}(\mathcal{A}_i)} \lambda_i^{EP}(r_i^{EP}). \quad (2)$$

In other words, AS  $i$ 's routing decision for customer-reachable destinations depend only on the routing decisions for its other customer-reachable destinations, and are independent of the routing decisions for its peer-provider-reachable destinations. Similarly, AS  $i$ 's routing decisions for its peer-provider-reachable destinations are independent of that of its customer-reachable destinations. When the routing decisions of AS  $i$  are decomposed for customer- and peer-provider-reachable destinations, we say that it follows *the standard joint-route preference policy*.

We now show the pleasant but surprising result that egress route selection for interdomain traffic engineering in the current Internet is stable. In order to do so, we note that there exist two BGP subsystems in the network. The first BGP subsystem is  $S_C = (G, \text{pt}, \sigma, \mathbb{D}^C, \tilde{\mathbb{P}}, \mathbb{P}^C)$ , where  $\mathcal{D}_i^C$  is the set of customer-reachable destinations for AS  $i$ , and  $\mathcal{P}_i^C = \cup_{d \in \mathcal{D}_i^C} R_{i \rightarrow d}^C$  is the set of all customer routes of AS  $i$ . The second BGP subsystem is  $S_{EP} = (G, \text{pt}, \sigma, D, \tilde{\mathbb{P}}, \tilde{\mathbb{P}})$ . It is easy to see that  $S_C$  is self-contained. Given any stable route selection  $\hat{r}^C$  for  $S_C$ ,  $S_{EP}|_{r^C = \hat{r}^C}$  is also self-contained. Therefore, we can establish the BGP protocol convergence for the whole network through sequential composition of these two self-contained BGP subsystems:

**Theorem 4** *The network has a unique stable route selection which BGP is guaranteed to converge to, and is guaranteed to be robust, if the following conditions hold:*

1. *there is no provider-customer loop in the network;*
2. *all ASes have fixed typical export policies;*
3. *the routing decisions for customer-reachable and peer-provider-reachable destinations follow the standard joint-route preference policy.*

**Proof:** We shall prove by sequential composition of two self-contained BGP subsystems that the network has a stable network route selection which BGP is guaranteed to converge to, and that the network is guaranteed to be stable. For proof of uniqueness of the stable network route selection, please refer to the proof in [55].

Let  $\tilde{\mathcal{P}}_i$  be the set of all possible paths for AS  $i$ . The first BGP subsystem we consider is  $S_C = (G, \text{pt}, \sigma, \mathbb{D}^C, \tilde{\mathbb{P}}, \mathbb{P}^C)$ , where  $\mathcal{D}_i^C$  is the set of customer-reachable destinations for AS  $i$ , and  $\mathcal{P}_i^C = \cup_{d \in \mathcal{D}_i^C} R_{i \rightarrow d}^C$  is the set of all customer routes of AS  $i$ .

The BGP subsystem  $S_C$  is self-contained. Consider an arbitrary AS  $i$  and an arbitrary  $d \in \mathcal{D}_i^C$ . By definition of  $\mathcal{D}_i^C$ , there exists at least one customer route  $P = (v_k, v_{k-1}, \dots, v_0)$  with  $v_k = i$  and  $v_0 = d$ , where each link  $(v_i, v_{i-1})$  is a provider-customer link, for  $i = k, k-1, \dots, 1$ . Initially, AS  $d$  has a trivial customer route  $(d)$  to itself. Since each AS prefers customer routes strictly over peer/provider routes, it can be shown by induction that AS  $i$  eventually will get a customer route to  $d$ .

There is no P-cycle in the P-graph of  $S_C$ . Suppose for the sake of contradiction that there is a P-cycle. We will show that there is a  $PC$ -loop in this case. Since the two partial route profiles connected by an improvement edge are of the same AS, it suffices to consider the sub-path edges on a P-cycle. Consider an arbitrary sub-path edge on the P-cycle from a partial route profile  $\tilde{r}_u^k$  to  $\hat{r}_v^l$ . AS  $v$  must be a customer of  $u$ , because any link on a customer route is a provider-customer link. Thus if we follow the P-cycle and examine all the sub-path edges along our way, we will get a  $PC$ -loop, which is a contradiction.

By Corollary 3, BGP protocol will converge on  $S_C$ . Thus each AS  $i$  will have a stable partial route profile to destinations in  $\mathcal{D}_i^C$ .

Denote by  $\hat{r}^C$  any stable route selection for  $S_C$ . The second BGP subsystem we consider is  $S_{EP} = (G, \text{pt}, \sigma, D, \tilde{\mathbb{P}}, \mathbb{P})$ . It is easy to see that  $S_{EP}|_{r^C = \hat{r}^C}$  is trivially self-contained for any stable route selection  $\hat{r}^C$ .

We shall prove that the P-graph of  $S_{EP}|_{r^C = \hat{r}^C}$  does not contain a P-cycle. Suppose for the sake of contradiction that there is a P-cycle. We will show that there is a  $PC$ -loop in this case. Again, it suffices to consider the sub-path edges on the P-cycle. Consider an arbitrary sub-path edge on the P-cycle from a partial route profile  $\tilde{r}_u^k$  to  $\hat{r}_v^l$ .

We first note the fact that  $\tilde{r}_v^l$  cannot be AS  $v$ 's partial route profile to customer-reachable destinations. Otherwise, by applying similar argument as for  $S_C$ , we can show that all sub-path edges on the P-cycle are from a provider to a customer, which contradicts the assumption that there is no  $PC$ -loop. This fact also implies that  $v$  cannot be a peer  $u$ , because if  $\tilde{r}_u^k(d)$  is a peer route for  $u$ , the sub-path  $\hat{r}_v^l(d)$  must be a customer route for  $v$ . Thus  $v$  can only be a provider of  $u$ . If we follow

the P-graph and examine all the sub-path edges along our way, we will get a  $PC$ -loop, which is a contradiction.

By Corollary 3, BGP protocol will converge on  $S_{EP}|_{r^C=\hat{r}^C}$  for any stable  $\hat{r}^C$ . Thus each AS  $i$  will have a stable partial route profile to destinations in  $\mathcal{D}_i$ . But in this case, a partial route profile to  $\mathcal{D}_i$  is exactly the complete route profile for  $i$ . Thus we have shown that BGP protocol converges for the whole network on all destinations.

In addition, it is not hard to see that the above proof holds even with arbitrary link/node failures, thus the network is robust. ■

Note that in the preceding theorem we require that customer routes are strictly preferred over peer routes; *i.e.*, customer routes  $\succ$  peer routes. One might suspect that the above theorem still holds if customer routes  $\succeq$  peer routes. However, Figure 3 gives a counter example and shows that there exists no stable route selection in this case.

#### 4.1 Stability with Multihomed Stub ASes Adopting Smart Routing Algorithms

As an application of Theorem 4, next we show that the recent trend of using smart routing to select egress routes does not introduce routing instability. Specifically, in [26], Goldenberg *et al.* propose algorithms to coordinate the egress route selection for multiple destinations to optimize performance under cost constraint. Using simulations, they show that their algorithms do not introduce instability. Below, we show that given that the conditions stated in Theorem 4 are satisfied, the conditions still hold when multihomed stub ASes adopt smart routing algorithms; thus, such algorithms do not introduce instability. First, adopting smart routing algorithms does not change the network topology; therefore, the first condition still holds. Second, adopting smart routing algorithms does not change the export policies. Third, a multihomed stub AS has only providers; therefore, its routing decisions, although coordinated, are inherently decomposed. Last, a multihomed stub AS follows the joint-route preference policy since it has only provider-routes to reach other destinations. To summarize, all of the conditions still hold when multihomed stub ASes adopt smart routing algorithms. Therefore, multihomed stub ASes adopting smart routing algorithms do not introduce routing instability.

## 5 Measurement and Simulation Studies of Egress Route Selection for Interdomain Traffic Engineering

The preceding sections analyze the stability of route selection for interdomain traffic engineering and prove that convergence and uniqueness of route selection can be guaranteed when there is no provider-customer loop, and all ASes follow the typical export policy and standard joint-route preference policy.

In this section, we complement the preceding analysis by investigating 1) the extent to which current Internet route selection satisfies the policies; and 2) the likelihood of instability when the policies and no- $PC$ -loop condition are violated.

## 5.1 Methodology

We first present our methodology. Specifically, we derive a necessary and sufficient condition to uniquely determine provider-customer relationships. We then use this condition to infer Internet topology. Simulation setup is also described in this section.

### 5.1.1 Inferring AS topology

We construct an Internet AS topology from multiple vantage points by using the aggregated BGP tables of Routeviews [46] and Looking Glass servers [36]. Specifically, we remove prepended AS numbers from the AS paths in the BGP table and filter out the paths with loops. We then construct an undirected AS-level topology graph as follows. Each AS has a unique node in the graph, and there exists an edge between two AS nodes if they ever appear in pair in an observed BGP route. The edges in this graph represent the connectivity among ASes.

We next infer business relationships among ASes to produce the *AS business-relationship graph*, denoted by  $G_b$ . Our inference of  $G_b$  consists of three steps. Firstly, we take the approach in [48] to infer peer relationships. Secondly, we infer provider and customer relationships for the remaining edges. Lastly, we remove edges with unknown relationships and label the remaining edges with the inferred relationships accordingly. In particular, in the second step, we construct a business-relationship inference graph, denoted by  $G_{infer}$ , to infer provider-customer relationships. In [3], Battista *et al.* map the inference of provider and customer relationships as a 2SAT problem. However, their method infers just one satisfiable solution. Thus, when the inferred business relationship between a pair of neighboring ASes is different from verification, it is unknown whether the error is due to ambiguity (*i.e.*, non-unique solutions) or model error. To overcome this problem, we construct a business-relationship inference graph as follows. Each pair of neighboring ASes,  $i$  and  $j$ , has two corresponding vertices in  $G_{infer}$ :  $v_{ij}$  and  $v_{ji}$ , where the vertex  $v_{ij}$  represents that  $i$  is a provider of  $j$ , while  $v_{ji}$  represents that  $j$  is a provider of  $i$ . We say that  $v_{ij}$  and  $v_{ji}$  are mirrors of each other. There exist edges between  $v_{ij}$  and  $v_{jk}$  in  $G_{infer}$  if and only if  $(i, j, k)$  or  $(k, j, i)$  appears as a segment of an observed route. In other words, from each route, we take all 3-tuple segments  $(i, j, k)$  and add two directed edges to the inference graph: one is from  $v_{ij}$  to  $v_{jk}$ , and the other from  $v_{kj}$  to  $v_{ji}$ . The directed edge from  $v_{ij}$  to  $v_{jk}$  encodes the fact that if  $i$  is a provider of  $j$  and  $(i, j, k)$  appears as a route segment,  $j$  must be a provider of  $k$  because of the no valley constraint. Given this construction and applying the result in [2], we have the following necessary and sufficient condition to check if the business relationship between a pair of neighboring ASes is uniquely determined:

**Theorem 5** *If all routes are valley-free, and ASes have only provider-customer relationships, then AS  $i$  is a provider of  $j$  if and only if in  $G_{infer}$ , vertex  $v_{ij}$  has a path to its mirror vertex  $v_{ji}$  and  $v_{ij}$  has no path back to  $v_{ji}$ .*

We apply the preceding theorem on  $G_{infer}$  to infer provider and customer relationships. We find that 85% of AS relationships can be uniquely determined. In order to validate our inference results, we compare the set of inferred customers of AT&T using our approach with that using the approach in [23], where Gao verified with AT&T that 96.3% of AT&T-related relationships were

correctly inferred. Our comparison shows that 98.8% of our inferred relationships are consistent with those using Gao’s approach. We further validate our results by conducting email surveys with randomly selected regional transit ISPs. The results of the surveys show that all the inferred provider-customer relationships are correct.

In order to make the simulations more efficient, we iteratively remove 6157 single-homed ASes whose route selection will not affect that of others. The remaining AS graph, denoted by  $G'_s$ , has 13,048 ASes and 37,999 links and is used in our simulations.

We observe that the inferred network topology  $G'_s$  has about 1.3% of ASes involved in  $PC$ -loops. We further find that  $PC$ -loops are introduced because some customers carelessly provide transit services for their providers, and these customers are inferred as providers as a result. Note that in this section  $PC$ -loops are not defined by the real AS business relationships; instead, they are defined by the business relationships inferred from observed routes determined by the export policies. Note also that the existence of  $PC$ -loops does not invalidate Theorem 5 since the aggregated BGP table used to construct  $G_{inferred}$  is not complete; therefore, the business relations of each link along a  $PC$ -loop may still be uniquely determined by applying Theorem 5.

To remove the  $PC$ -loops, we take into account the common belief that providers typically have more neighbors than their customers. Specifically, we first locate all the  $PC$ -loops in the graph. Then, for each  $PC$ -loop, we compute for each link along the loop the ratio of the provider’s degree and the customer’s degree, and iteratively remove the link with the lowest ratio, until there is no  $PC$ -loop. We denote by  $G_s$  the induced subgraph of  $G'_s$  after breaking all  $PC$ -loops.  $G'_s$  is only used to evaluate the impact of  $PC$ -loop on routing stability through simulation, and  $G_s$  is used in all other simulations.

### 5.1.2 Simulation setup

An important component of our simulation studies is route ranking tables. For AS  $i$  who does not coordinate the route selection of multiple destinations, we use the subjective routing framework to construct its route ranking table [12]. The subjective routing framework is motivated by the observation that different ASes often use different performance metrics in comparing routes. Thus, in this framework, there is a set  $M$  of performance metrics assigned to each link. Each AS computes the cost of a route using its own set of weights. Specifically, AS  $i$  has a set of weights,  $W_i = \{w_{i,m} | m \in M\}$ , where  $w_{i,m}$  is the weight associated with the performance metric  $m$ . Note that  $w_{i,m} = 0$  if  $i$  is not concerned with the metric  $m$ . Let  $C_l^{(m)}$  be the value of metric  $m$  at link  $l$ . Given a route  $r_{i \rightarrow d}$  from AS  $i$  to destination  $d$ , AS  $i$  computes the cost of this route as  $c(r_{i \rightarrow d}) = \sum_{m \in M} w_{i,m} \sum_{l \in r_{i \rightarrow d}} C_l^{(m)}$ . For each destination, AS  $i$  chooses the route with the lowest subjective cost as its best route for that destination.

For an AS  $i$  who coordinates its route selection of multiple destinations, we construct its ranking table as follows. First, for each destination  $d$ , we compute the set  $R_{i \rightarrow d}$  of all feasible valley-free routes from  $i$  to  $d$  in  $G_s$ , assuming all ASes have typical export policies. Then we construct the set of all possible route profiles  $R_i = \prod_{d \in \mathcal{D}} R_{i \rightarrow d}$ . For efficiency, we do not explicitly store  $R_i$ ; instead, we store just the set of all feasible routes to all destinations (*i.e.*,  $\cup_{d \in \mathcal{D}} R_{i \rightarrow d}$ ), and assign a unique ID to each route in this set; therefore, we represent a route profile using a set of IDs

corresponding to the routes in the route profile. Finally, we construct the ranking table of AS  $i$  by randomly permuting the entries of  $R_i$ .

We implement our own event-driven simulator to study the stable route selection problem for interdomain traffic engineering. It simulates BGP protocol process such as route import/export, route announcement/withdrawal, and so on. Each AS selects its routes as described above. We also add random delays to route import/export events in order to simulate network asynchronousness. In each experiment, we randomly choose a set of ASes as destinations, and all other ASes exchange routes to these destinations.

To detect instability, for each AS, our simulator keeps a history of its selected route profiles. Specifically, according to its route selection history, each AS constructs a directed stability graph with each node representing a unique route profile and each directed edge representing a temporal transition between two route profiles. An AS has no stable route selection if all nodes of the stability graph are in one single strongly connected component. Hereafter, we refer to such ASes as *unstable* ASes. Since this condition is a sufficient condition, we may underestimate the extent of instability. In order to avoid taking initial route exchanges as unstable route selection, we wait for a long enough time before checking instability. Specifically, we start to keep a history of previous best route profiles for each AS after 500 simulation steps when all ASes have routes to all destinations. We start to check the instability condition for each AS every 20 simulation steps after the routing history starts. We run the simulation for 7,000 simulation steps so that the number of ASes identified as unstable does not change any more, and take this number as the number of unstable ASes.

## 5.2 Route Selection Practice in the Current Internet

We start with an investigation on the route selection practice of the current Internet. Since the interaction of multiple destinations can cause instability, as we pointed out in Section 3.3, it is important to study the extent to which ISP route selection satisfies the policies proposed in Theorem 4, which can guarantee the existence of a unique stable route selection. In particular, we investigate the extent to which 1) the standard individual-route preference policy is followed, and 2) the standard joint-route preference policy is followed, in the current Internet.

### 5.2.1 The standard individual-route preference policy

We first investigate the extent to which ISP route selection follows the standard individual-route preference policy. For each AS under study, we extract its available routes and the local preference value it assigned to each route from its BGP routing tables. We label each extracted route as a customer, a peer, or a provider route, using our inferred provider-customer and peer business relationships. According to the standard BGP routing decision process, a route is strictly preferred over another one if the route has a higher local preference value. Based on this rule, we compare the local preference values of two routes of the same prefix. For each prefix, we check three types of violations: a peer route has an equal or larger local preference value than a customer route (*i.e.*,  $E \succeq C$ ); a provider route has an equal or larger local preference value than a peer route (*i.e.*,  $P \succeq E$ ); a provider route has an equal or larger local preference value than a customer route

(i.e.,  $P \succeq C$ ). If a prefix has a violation, then we refer to the prefix as a violating prefix and the involved route as a violating route. We count the total number of violating prefixes and compute the percentage of violating prefixes for each AS under study. Similarly, we compute the percentage of violating routes as well. Table 1 summarizes the results.

We observe that 11 out of the 18 ASes do not have any violations of the standard individual-route preference policy in their route selection. Three of them do not completely follow the preference order but the percentage of violations is small, on the order of 0.02% to 0.06%. However, for the remaining four ASes, the percentage of violations can be quite high, from about 1.1% to as high as 2.4%. Given the vast number of destination prefixes in the Internet (each AS under study has about 150K destination prefixes), such percentage translates to a large number of destination prefixes. To further investigate the reasons of violations, we divide the violations into three categories as described above:  $E \succeq C$ ,  $P \succeq E$ , and  $P \succeq C$ . Then, for each of these three categories, we compute the percentages of violating prefixes and routes. We observe that the majority of the violations of the preference order come from route preferences violating preferring peers over providers (i.e.,  $P \succeq E$ ), which indicates potential load balancing considerations. The violations of the other two categories are negligible and therefore, they are not listed in the table. Although we sample only a small fraction of ASes in the current Internet, we believe that the sampled ASes are representative, as our results are also consistent with those in [54]. However, [54] studied only the percentage of prefixes that have routes violating the standard individual-route preference policy, while we also investigate the percentage of routes with violations and where the violations come from and their percentages. In addition, we observe smaller percentages of violations compared with that in [54]; for instance, AS 5511 has only 1.9%, instead of 3.5% in [54], of prefixes violating the policy.

In summary, although the route selection of most of the ASes satisfies the standard individual-route preference policy (11 with no violations and 3 with almost negligible violations), the route selection of many ASes (4 out of 18  $\approx 22\%$ ) does not. When a large number of ASes do not follow the preference order, their preferences might interact and cause instability, as we have shown in Section 3.

### 5.2.2 The standard joint-route preference policy

Next we investigate the extent to which AS route selection follows the joint-route preference policy. Since there is no direct way to check if the route selection of an AS follows the standard joint-route preference policy, we derive the following condition for checking violations of the policy.

Consider a specific AS  $i$ . Assume we take two snapshots of the available routes of AS  $i$  at times  $t$  and  $t'$ , where  $t < t'$ . Let  $R_i(t)$  and  $R_i(t')$  be the sets of available route profiles to the AS (from its routing caches) at these two snapshots. Let  $r_i(t)$  be the route profile that  $i$  chooses at time  $t$ . Let  $r_i^C(t)$  and  $r_i^{EP}(t)$  be the subsets of the routes in  $r_i(t)$  for customer-reachable, and peer-provider-reachable destinations, respectively. We can similarly define  $r_i(t')$ ,  $r_i^C(t')$ , and  $r_i^{EP}(t')$ . A scenario that the route selection of customer-reachable destinations is not decoupled from that of peer-provider-reachable destinations is that  $r_i^C(t) \in R_i^C(t')$ ,  $r_i^C(t') \in R_i^C(t)$ ,  $r_i^{EP}(t) \neq r_i^{EP}(t')$ , and  $r_i^C(t) \neq r_i^C(t')$ . In other words, if route selection for customer-reachable and peer-provider-reachable destinations is decoupled, then a change of routes for peer-provider-reachable destina-

ASN	Degree	% of prefixes		% of routes	
		Total	$P \succeq E$	Total	$P \succeq E$
553	131	0	0	0	0
852	111	0	0	0	0
3257	264	0	0	0	0
5388	112	0	0	0	0
5713	23	0	0	0	0
6730	516	0	0	0	0
7018	1964	0	0	0	0
7474	129	0	0	0	0
9132	306	0	0	0	0
15837	130	0	0	0	0
17233	8	0	0	0	0
6539	274	0.02	0.01	0.01	0.004
3561	610	0.03	0.02	0.01	0.009
6667	394	0.06	0.04	0.04	0.03
3549	669	1.14	1.10	0.14	0.138
7911	287	1.54	1.40	0.26	0.23
5511	190	1.90	1.90	0.63	0.63
8220	764	2.41	1.84	0.76	0.56

Table 1: Route preference violating the policy of customer  $\succ$  peer  $\succ$  provider. A major source of violations comes from not strictly preferring peer routes over provider routes ( $P \succeq E$ ).

tions (namely  $r_i^{EP}(t) \rightarrow r_i^{EP}(t')$ ) should not cause any route change of customer-reachable destinations (namely  $r_i^C(t) \rightarrow r_i^C(t')$ ). Similarly, we can define sufficient conditions for other cases of the violations of the standard joint-route preference policy.

Checking the condition defined above requires us to compare across two snapshots of an AS’s routing tables. One potential problem is that an AS may change its preference between these two snapshots and thus make the comparison invalid. To overcome the problem, we select two snapshots when they are close in time and satisfy a consistency check. Specifically, the consistency check we adopt is that if AS  $i$  chooses route profile  $r_i$  over  $r_i'$  when presented with both profiles at time  $t$ , then if both profiles are available at time  $t' > t$ , then the AS should not choose  $r_i'$  over  $r_i$ . Given this consistency check, we choose snapshots in the following way. We first dump the complete BGP routing tables from the chosen Looking Glass servers every 3 minutes for a period of 30 minutes, starting at random times. We then select the sequences of snapshots which do not contain any inconsistency.

Our result shows that the route selection of most ASes does not violate the sufficient conditions of the standard joint-route preference policy. However, we observe that some ASes violate this decoupling condition for some of their prefixes. For example, we observe that at least 150 prefixes whose route selection violates the decoupling condition.

### 5.2.3 Summary

Our measurements of the route selection in the current Internet show that the standard individual- and joint-route preference policies are largely satisfied. This could be one of the reasons for the stability of the current Internet. However, we also observe many instances of violations. For example, we observe that the route selection of 22% of the ASes does not satisfy the standard individual-route preference policy. The existence of a non-negligible percentage of ASes whose

route selection does not satisfy the policies could be alarming, as we will see in our simulations in the next subsection that even when just a small number of ASes conduct route selection for interdomain traffic engineering but their route selection does not satisfies the standard joint-route preference policy, instability could occur.

### 5.3 Routing instability when each destination is routed separately

We start our study of routing instability when no AS coordinates its route selection. Although the focus of this paper is on routing instability caused by coordination of route selection, since there is no previous simulation study on the single destination case, we conduct the first set of experiments as reference points. In our simulations, we randomly choose a destination AS that originates route announcements. The remaining ASes follow BGP protocol process to select the best route with the minimum subjective cost to the chosen destination.

Our first experiment uses the topology with *PC*-loops, *i.e.*,  $G_s$ , to study routing instability. In this experiment, all ASes have typical export policies, and strictly follow the standard individual-route preference. However, due to the existence of *PC*-loops, we still observe unstable ASes. Figure 8(a) shows the empirical cumulative distribution of the number of unstable ASes obtained from our experiments. We also conduct a distribution fitting and find that the extreme value distribution best fits the empirical one. Figure 8(b) also plots their density functions. To confirm that it is *PC*-loops that causes instability, we repeat the same experiment using  $G_s$ , where all *PC*-loops are removed, and we do not observe any instability in simulations.

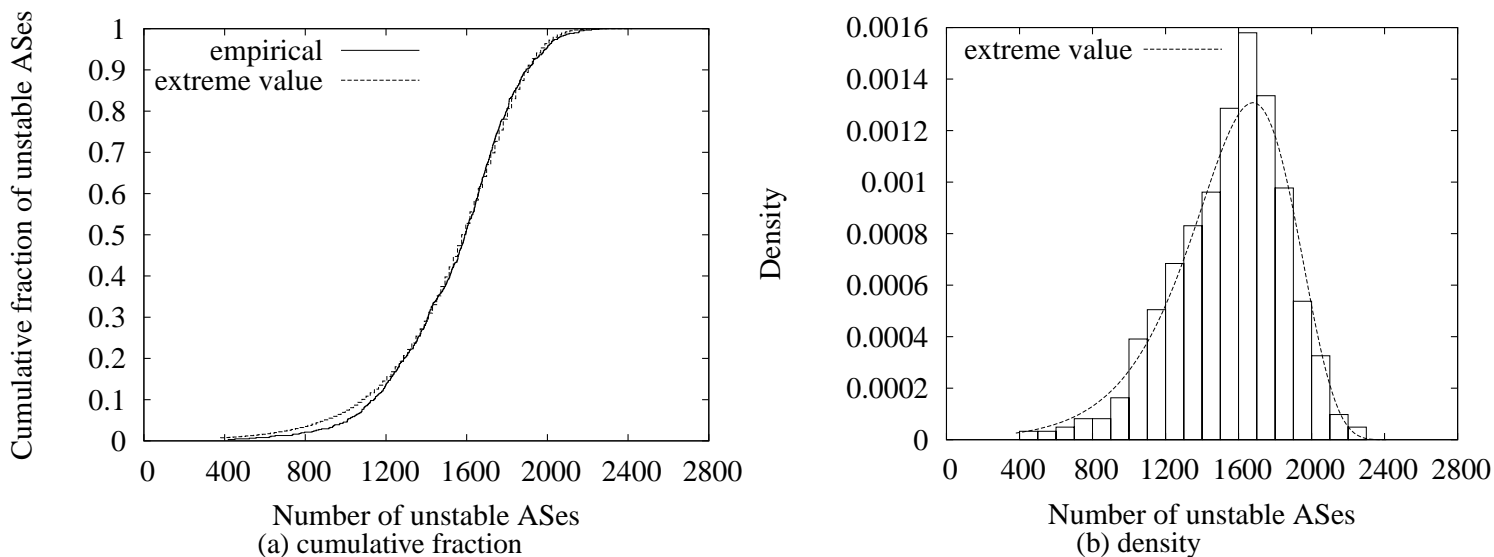


Figure 8: Distribution of total number of unstable ASes due to *PC*-loops.

Our second experiment uses the *PC*-loop-free topology,  $G_s$ , to study routing instability when ASes violate the standard individual-route preference. In this experiment, ASes have typical export policies. Each AS violates the standard individual-route preference with probability  $p_v = 0.03$ ; for

instance, with both a customer route and a peer route to a destination, an AS chooses the peer route instead of the customer route with probability 0.03. This probability is chosen because we observe that at most 3% of prefixes have routes violating the standard individual-route preference in the current Internet [55]. In order to study the impact of the violation probability on the number of unstable ASes, we also repeat the experiment with doubled violation probability  $p_v = 0.06$ .

Figure 9(a) shows the empirical cumulative distributions for both experiments. Similarly, we conduct a distribution fitting and find that the negative binomial distribution best fits them. We also plot in Figure 9(b) the density functions of both distributions for the case where  $p_v = 0.03$ . We observe that the number of unstable ASes increases when  $p_v$  is doubled. In particular, we find that on average, there are 43 unstable ASes when  $p_v = 0.03$ ; when the violation probability is doubled, the average number of unstable ASes is more than doubled to 95. Comparing this experiment with the preceding one, we also observe that violation of the topological condition is more likely to lead to routing instability.

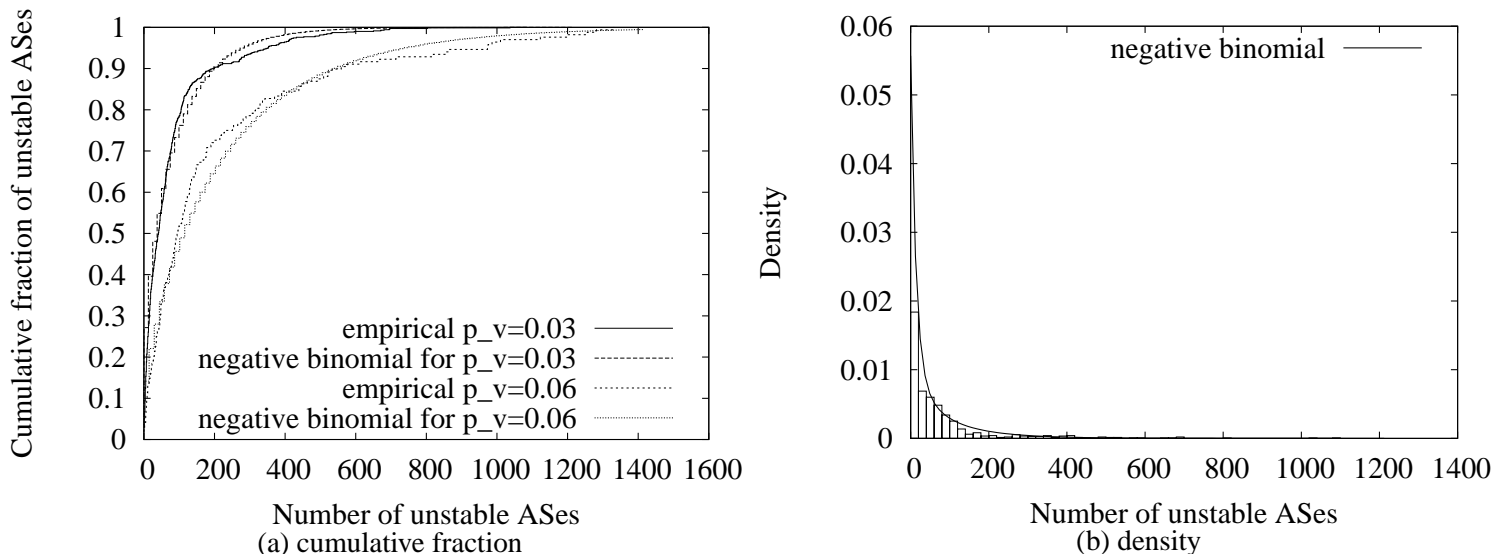


Figure 9: Distribution of total number of unstable ASes due to violation of the standard individual-route preference.

## 5.4 Routing instability caused by route coordination

Finally, we investigate routing instability caused by coordinated route selection of multiple destinations.

We start with a candidate set consisting of a randomly chosen Tier-2 AS. We then randomly choose the neighboring ASes of the candidates with probability 0.5 as the ASes that coordinate their route selections, and add them to the candidate set. This process continues until the set consists of enough number of ASes. We choose the candidate ASes in this way to model a scenario where ASes are more likely to coordinate route selections when their neighbors are doing so.

We also limit our choice of candidate ASes to Tier-2 and Tier-3 ASes since Tier-1 ISPs are very cautious and less likely to actively coordinate their routes to achieve some traffic engineering objectives. To investigate the potential seriousness of the problem, we setup the experiments so that only 40 ASes coordinate route selection for only 2 destinations and violate the standard joint-route preference policy. All remaining ASes select routes for each destination separately.

We study the following two cases: (a) the remaining ASes strictly follow the standard individual-route preference; and (b) the remaining ASes violate the standard individual-route preference with probability 0.03. Figure 10 shows the empirical distribution of the number of unstable candidate ASes for both cases. We conduct a distribution fitting and find that the negative binomial distribution best fits the empirical distributions, as shown in the figures. We observe in case (a) that in worst cases, almost all 40 candidate ASes are unstable in the network. This result is surprising in that 40 ASes consist of a very small percentage (40 out of 13048) of the total number of ASes. Furthermore, 2 destinations are not many destinations. We also vary the number of ASes who coordinate route selection and the number of destinations. We observe that the number of unstable ASes further increases as the number of ASes who coordinate route selection but do not follow the joint-route preference policy increases. We also observe in case (b) that the number of unstable ASes strictly increases when the remaining ASes violate the standard individual-route preference.

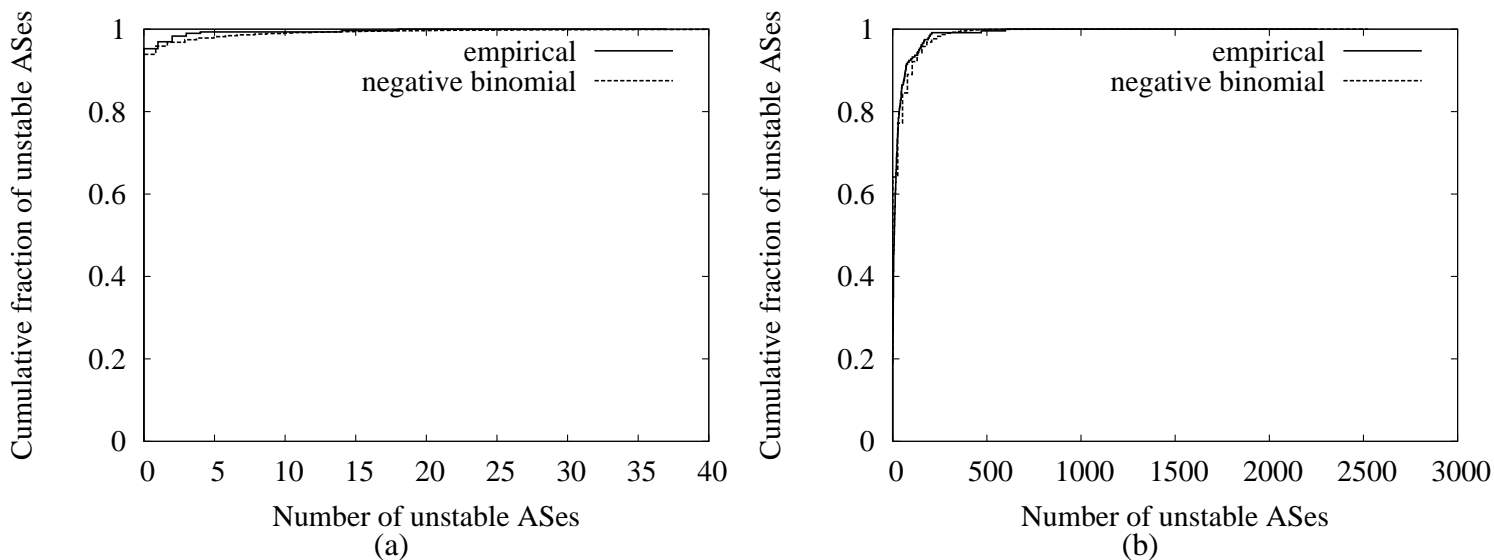


Figure 10: Distributions of total number of unstable ASes due to violation of the standard joint-route preference policy, when the remaining ASes that do not coordinate route selections and either (a) strictly follow or (b) violate with probability 0.03 the standard individual-route preference.

## 6 Route Selection for General Interdomain Traffic Engineering

In the preceding sections, the preference of an AS depends only on egress route profiles and is independent of ingress traffic demand patterns. As a result of this independency, we derive a set of practical guidelines which can guarantee stability for egress interdomain traffic engineering. This independency is justified when the traffic demands of an AS to its destinations are known, and thus can be considered as constants. Specifically, these demands will be used as constant parameters in the ranking function of an AS to determine the relative ranking of route profiles. Conceptually, therefore, these demands do not need to appear explicitly in the route ranking table as conditions. For example, in Figure 1, the traffic demands to  $D_1$  and  $D_2$  will be used in determining the relative ranking of route profiles of  $S$ . However, these traffic demands do not need to appear explicitly in the route ranking table. Some examples where this is true include multihomed stub ASes, and ISPs whose aggregated traffic to its major destinations is relatively stable.

However, in a more general case, the preference of an AS could include both egress route profiles and ingress traffic patterns. We call the stability problem under this model the *stable route selection for general interdomain traffic engineering problem*. Route selection for general interdomain traffic engineering is likely to be important for an intermediate transit ISP whose ingress traffic varies substantially with its own route selection. The objective of this section, therefore, is to investigate the stability of route selection for general interdomain traffic engineering.

### 6.1 Motivation

#### 6.1.1 A Motivating Example

We start with an example to show that ASes may adopt general local policies. The example also shows that with inbound-dependency, whether or not a network is stable can depend on the route selection algorithms.

From our email surveys, it is clear that inbound-dependent route selection is important for a transit ISP whose inbound traffic varies substantially with its own route selection. Figure 11 shows an example network which is motivated by the increasing usage of multihoming and its potential effects on some transit ISPs. The example network is constructed in such a way that it satisfies all conditions to guarantee stability for inbound-independent route selection [22]: there is no provider-customer loop in the network; each AS follows the *typical export policy*; and an AS prefers customer routes over provider routes. The example network avoids peering links to have a clean setup.

A special feature of this example network, however, is that the ranking of egress routes at  $B$ , who is one of the two competing transit providers of source  $S$ , depends on its inbound traffic. For generality, we say that  $B$  ranks *outcomes*, instead of just egress routes. An outcome consists of both an egress route and ingress traffic pattern. For generality, we assume a ranking table at each AS, which lists, in decreasing order, all of the potential outcomes. Note that in practice, a ranking table can be implemented, compactly, by an objective or utility function. Specifically,  $\{S\}BFD$  denotes the outcome that  $B$  uses the egress route  $BFD$  and  $S$  sends traffic for destination  $D$

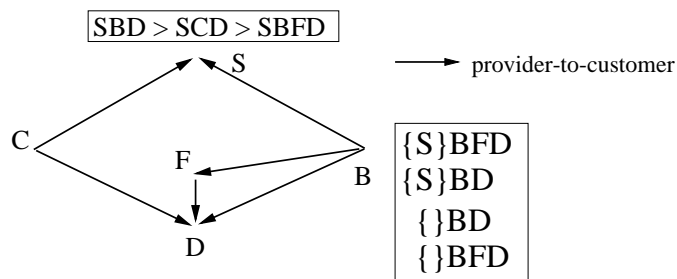


Figure 11: The ranking of egress routes at  $B$  depends on inbound traffic  $c$ .  $S$  is the source, and  $D$  is the destination.

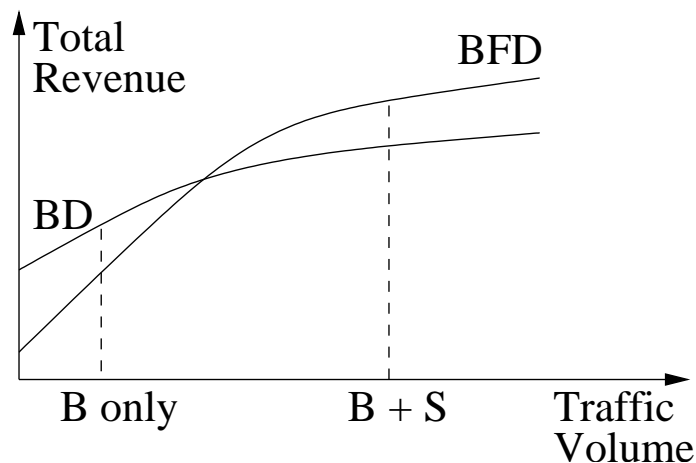


Figure 12: A revenue function justifying the route selection behavior of  $B$  in Figure 11. “ $B$  only” denotes the traffic volume when  $S$  does not use  $B$  as its transit provider; and  $B + S$  denotes that when  $S$  uses  $B$ .

through  $B$ ;  $\{\}BD$  denotes the outcome that  $B$  uses the route  $BD$  and  $S$  does not send any traffic through  $B$ .

This example network does not appear to be a pathological case and can well happen in practice.  $S$  is a multihomed network with two providers  $C$  and  $B$  to improve reliability. The ranking table of  $S$  is constructed according to the standard BGP decision process:  $S$  prefers routes with small AS-hop counts; for two routes with the same AS-hop count, it uses the next-hop ID to break the tie. As for  $B$ , when traffic volume is high (*i.e.*, when  $S$  uses  $B$  as its transit provider),  $B$  selects  $BFD$  over  $BD$ ; on the other hand, when traffic volume is low (*i.e.*, when  $S$  does not use  $B$  as its transit provider),  $B$  chooses  $BD$  over  $BFD$ . A potential revenue function that may cause this scenario to happen is shown in Figure 12; that is,  $BFD$  is more profitable for  $B$  when the traffic volume is high, while  $BD$  is more profitable for  $B$  when the traffic volume is low. Note that it is possible to reverse the provider-customer relationship of the AS pairs,  $CD$ ,  $FD$ ,  $BF$ , and  $BD$ . Then the preference of  $B$  can be justified by cost instead of revenue.

### 6.1.2 Instability of a Traffic-Demand-Matrix-Based Route Selection Scheme

A common approach for  $B$  to implement inbound-dependent route selection is to use a traffic-demand-matrix-based algorithm (e.g., [5, 26]). The basic structure of such an algorithm is that time is divided into multiple periods. During each time period, the algorithm measures the traffic demand matrix. At the end of each time period, the algorithm computes and installs the optimal route selection for the next period.

Specifically,  $B$  would implement a route selection algorithm as follows. During each time period  $n$ ,  $B$  estimates total traffic demand to destination  $D$ ; At the end of time period  $n$ ,  $B$  computes the optimal route selection ( $BFD$  or  $BD$ ), based on the measured inbound traffic demand and its traffic engineering objectives.  $B$  then installs the optimal route selection at the beginning of time period  $n + 1$ . As we have discussed in the introduction, this algorithm can be implemented either by a network operator manually, which will operate at a longer time scale, or by a traffic engineering program, which will operate at a much faster speed.

Given the above route selection algorithm, assume that  $B$  initially chooses egress route  $BD$ .  $B$  exports  $BD$  to  $S$ ; therefore,  $S$  chooses  $SBD$  over  $SCD$ , and the traffic from  $S$  to  $D$  goes through  $B$ . However, given this high inbound traffic demand,  $B$  prefers  $BFD$  over  $BD$ ; thus  $B$  switches its route selection to  $BFD$  and exports to  $S$ . This change of egress route causes  $S$  to choose  $SCD$  over  $SBD$ , and thus traffic of  $S$  no longer goes through  $B$ . Given that now the inbound traffic is low,  $B$  switches back to route selection  $BD$ , since it prefers  $BD$  over  $BFD$  at low traffic. Thus, we have obtained persistent route oscillations<sup>2</sup>.

### 6.1.3 Optimal and Stable Inbound-dependent Route Selection by a Single AS

The above instability is due to the fact that under the preceding traffic-demand-matrix-based route selection algorithm,  $B$  mis-associates the outcomes with its available actions (e.g.,  $B$  has two available actions in the preceding example: choosing  $BD$  or  $BFD$ ). To choose the optimal route and maintain stability, an AS  $i$  needs to correctly associate the outcomes with its actions; that is, the estimated inbound traffic pattern is a result of the chosen egress route. Learning the outcomes of all available egress routes, AS  $i$  chooses the optimal outcome. Figure 13 specifies a route selection algorithm which can guarantee stability and optimality, when only AS  $i$  adopts this inbound-dependent route selection algorithm. Note that in Figure 13,  $r_i$  is a route selection constructed from the routes exported by AS  $i$ 's neighbors. We refer to that a route profile  $r_i$  is overwhelmed by  $r'_i$  if (1) whenever  $r_i$  is available,  $r'_i$  is also available; and (2) choosing  $r'_i$  always yields strictly more preferable outcome than choosing  $r_i$ . This notion will be formalized in our general model of route selection algorithms.

Specifically, in the context of Internet interdomain route selection, when ASes are constrained by Internet business considerations, Theorem 6 shows that the algorithm in Figure 13 can guarantee stability and optimality. Due to space limitation, we omit its proof, and note that an induction proof can be constructed.

---

<sup>2</sup>This oscillation is different from that generated by classical single-path adaptive routing; for example, the classical routing scheme considers only latency [4].

```

▷  $T_{conv}$  is maximum time for routing convergence
▷  $T_m$  is the measurement time
▷  $R_i$  is the set of available, unoverwhelmed route selections
   constructed from routes exported by  $i$ 's neighbors
▷  $tm(r_i)$  represents the inbound traffic matrix when choosing  $r_i$ 

foreach  $r_i$  in  $R_i$ 
    install  $r_i$ 
    estimate  $tm(r_i)$  by
        waiting for  $T_{conv}$ 
        measuring  $tm(r_i)$  for  $T_m$ 
    if any route in  $R_i$  is overwhelmed
        remove it from  $R_i$ 

```

Figure 13: An inbound-dependent route selection algorithm.

**Theorem 6** *The network converges, and an AS  $i$  converges to its optimal outcome, if the following conditions are satisfied:*

1. *there is no provider-customer loop in the network;*
2. *all ASes except  $i$  adopt the typical export policy;*
3. *each AS prefers customer routes over peer/provider routes;*
4. *AS  $i$  adopts the route selection algorithm in Figure 13, and no other AS uses any inbound-dependent route selection.*

Consider the example in Figure 11. At the beginning,  $B$  does not know which of  $BD$  or  $BFD$  is the egress route to choose. So it can select either one. Later, it learns the outcomes of choosing  $BD$  and  $BFD$ . Since the outcome of choosing  $BD$  is more preferred than that of choosing  $BFD$ ,  $B$  chooses  $BD$ . For brevity, we also say that  $BD$  overwhelms  $BFD$ .

## 6.2 General Rational Route Selection Algorithms

It is clear from the preceding section that the stability of a network depends on not only the interaction of the local routing policies of the ASes in the network, but also the route selection algorithms implementing the policies. The instability studied by the previous studies is caused by policy interaction, while the instability identified in the preceding section is caused by the specific route selection algorithm. Since it is highly likely that more route selection algorithms will be designed, it is important to analyze the stability of a heterogeneous network where ASes run any reasonable route selection algorithms, not a homogeneous network where all ASes run a single, specific algorithm, for example, the greedy BGP algorithm, or the one in Figure 13. This is particularly

important in a network when ASes adopt different types of local policies (*e.g.*, some inbound independent while some dependent), since it is reasonable that then different ASes may choose route selection algorithms according to their local policies.

Below, we define the notion of *rational route selection algorithms*, and conduct our stability analysis using the general rational route selection algorithms. There are several advantages in conducting stability analysis based on the general notion of rational route selection algorithms. First, it allows us to establish stronger positive results in two senses: 1) it allows us to prove the stability of a heterogeneous network where different ASes can run different route selection algorithms, so long all of the algorithms are rational; 2) since the notion of a rational route selection algorithm is defined by its asymptotic behavior, if variations to a route selection algorithm do not change its asymptotic behavior (*e.g.*, non-persistent route dampening), the route selection algorithm is still rational, and thus the stability result still holds. Second, it allows us to establish stronger negative results; for example, if we show that a network is unstable under *any* rational route selection algorithms, it is stronger than to show that a network is unstable under a specific route selection algorithm.

The concept of rational route selection algorithms is motivated by previous work on adaptive learning [42] and learning on the Internet [20]. The models used in the previous game theoretical studies are normal form games. However, interdomain route selection is more of an extensive form game than a normal form game, since an intrinsic characteristic of interdomain route selection is that the available routes of an AS depend on those exported by its neighbors. In this paper, we shall explicitly model this dependency. In the sequel, we shall formalize our intuitive notion of rational route selection algorithms and explore the implications.

### 6.2.1 Rational Route Selection: Model

The notions of the network topology, paths, peering transformation, network route selection, route profile and combined route profiles are similarly defined as those in Section 3.2.

An intrinsic characteristic of path vector protocols such as BGP is that there are dependencies among route selections of ASes. Specifically, the route profiles available to  $i$  depend on the route advertisements it receives from its neighbors, which in turn depend on route selections of these neighbors. To capture this dependency, we define two operators  $C_i$  and  $A_i$  for each AS  $i$  as follows. For a set of paths  $\mathcal{P} \subseteq R$ , let

$$C_i(\mathcal{P}) = \{(i, j) \text{ pt}(i, j, P) | P \in \mathcal{P} \cap R_{j \rightarrow}\} \quad (3)$$

$$A_i(\mathcal{P}) = \{r_i \in \mathcal{R}_i | r_i(k) \in C_i(\mathcal{P}) \cup \{\epsilon\}, \forall k \in \mathcal{D}_i\} \quad (4)$$

Intuitively, if  $\mathcal{P}$  is the set of routes exported by  $i$ 's neighbors, then  $C_i(\mathcal{P})$  is the set of routes available to  $i$  in its routing cache, and  $A_i(\mathcal{P})$  is the set of route profiles that  $i$  can possibly choose from this routing cache. Note that AS  $i$  can always choose the empty path to any  $k \in \mathcal{D}_i$  regardless of  $C_i(\mathcal{P})$ .

The route selection objective of AS  $i$  (*i.e.*, its local preference) is represented by a utility function  $u_i(r_i, r_{-i})$ , which evaluates the payoff of the current network route selection  $r$  for  $i$ . Note that since we allow the utility of  $i$  to depend on not only  $i$ 's route, but also all other ASes' routes, it captures inbound-dependent route selection.

As is mentioned at the beginning of this section, we want to analyze the stability of a heterogeneous network where ASes run any reasonable route selection algorithms. In order to achieve this generality, we avoid any detailed specification of how the ASes actually select route profiles. Instead, we focus on the sequence of network route selections over time, and identify some general properties fulfilled by these sequences when ASes use any reasonable route selection algorithms that we consider.

We assume that there is a set of times  $T = \{0, 1, 2, \dots\}$  at which one or more ASes in the network change their route profiles. The elements of  $T$  should be viewed as the indices of the sequence of physical times at which these changes take place. At time  $t$ , the selected route profile of AS  $i$  is  $r_i[t]$ , and the network route selection is  $r[t] = (r_i[t])_{i \in V}$ . The sequence of network route selections is, therefore,  $\{r[t]\}_{t=0}^{\infty}$ .

Given a set  $H \subseteq \mathcal{R}$  of network route selections, we define the *projection* of  $H$  onto  $\mathcal{R}_i$  as

$$H_i = \{r_i \in \mathcal{R}_i \mid r \in H\}. \quad (5)$$

Accordingly, we define the *product* set  $H_{-i}$  as

$$H_{-i} = \{r_{-i} \in \mathcal{R}_{-i} \mid (r_{-i})_j \in H_j, \forall j \neq i\}. \quad (6)$$

The set  $H_{-i}$  represents all possible combined route profiles of all ASes except  $i$ , where AS  $j$ 's route profile is drawn from  $H_j$  for all  $j \neq i$ . Also, let

$$A_i(H_{-i}) = \bigcup_{r_{-i} \in H_{-i}} A_i(r_{-i}). \quad (7)$$

Recall that in the above definition,  $A_i(r_{-i})$  actually means  $A_i(\{(r_{-i})_j(k) \mid k \in \mathcal{D}_j, j \neq i\})$ .

Suppose that AS  $i$  has observed a set  $H$  of network route selections, and believes that each other AS  $j$  will select route profiles in  $H_j$ . It is reasonable, therefore, for  $i$  to believe that the route selections of the other ASes belong to the set  $H_{-i}$ , and that the route profiles possibly available to it will belong to the set  $A_i(H_{-i})$ . If there exist two route profiles  $r_i, r'_i \in A_i(H_{-i})$ , such that

1. whenever  $r_i$  is available,  $r'_i$  is also available;
2. choosing  $r'_i$  always yields strictly higher payoff than  $r_i$ ;

then it would be “unjustified” or “irrational” for  $i$  to choose  $r_i$ . Formalizing the above argument, we define the following operator  $U : 2^{\mathcal{R}} \mapsto 2^{\mathcal{R}}$ :

**Definition 1** Given  $H \subseteq \mathcal{R}$ , let

$$\begin{aligned} U_i(H) &= \{r_i \in A_i(H_{-i}) \mid \forall r'_i \in A_i(H_{-i}), P1 \vee P2, \\ &\text{where} \\ &\quad (P1) \exists r_{-i} \in H_{-i}, \text{ such that} \\ &\quad \quad r_i \in A_i(r_{-i}), r'_i \notin A_i(r_{-i}), \\ &\quad (P2) \exists r_{-i} \in H_{-i}, \text{ such that} \\ &\quad \quad r_i \in A_i(r_{-i}), r'_i \in A_i(r_{-i}), \\ &\quad \quad u_i(r_i, r_{-i}) \geq u_i(r'_i, r_{-i})\}, \\ U(H) &= \{r \in \mathcal{R} \mid r_i \in U_i(H)\}. \end{aligned}$$

The set  $U_i(H)$  is the set of route profiles that are not *overwhelmed* when each other AS  $j$  is limited to route profiles in  $H_j$ . If AS  $i$  believes that other ASes will select route profiles in  $H_{-i}$ , then it would be “unjustified” for AS  $i$  to choose any route profile not in  $U_i(H)$ , since every such route profile is guaranteed to be strictly worse than some other route profile in  $U_i(H)$ .  $U_i(H)$  thus formalizes our notion of the set of *unoverwhelmed* route profiles for AS  $i$  that are consistent with the route selections of other ASes  $H_{-i}$ .

Our intuitive notion of “rational route selection” is defined in terms of properties fulfilled by a sequence of network route selections.

**Definition 2**  $\{r_i[t] | t \in T\}$  is consistent with rational route selection if, for all  $t'$ , there exists  $t'' > t'$  such that for all  $t > t''$ ,  $r_i[t] \in U_i(\{r[s] | t' \leq s < t\})$ .  $\{r[t] | t \in T\}$  is consistent with rational route selection if each  $\{r_i[t] | t \in T\}$  has this property.

## 6.2.2 Example: BGP for Inbound-independent Interdomain Routing

The preceding definition of rational route selection is generic and does not specify how ASes actually select route profiles. Thus, it allows both centralized and distributed implementations. An example centralized implementation can be as follows. Each AS sends its utility function to a trusted third party. The third party then applies the operator  $U$  to compute for each AS a routing schedule (namely what route each AS should adopt at what time).

As an example of distributed implementation, below we analyze the *standard BGP route selection protocol* as it is used in interdomain route selection. By the standard BGP route selection protocol, we mean essentially the simple path vector protocol (SPVP) as defined in Fig. 5 of [29], extended to the case of joint multiple-destination route selection, and other features such as route dampening, so long some mild conditions are satisfied. We will show that the asymptotic best-response nature of BGP makes it a rational route selection algorithm, when the ranking of egress routes is independent of inbound traffic.

Specifically, we have the following result:

**Theorem 7** *The BGP protocol is consistent with rational route selection, if the following conditions are satisfied:*

- A1. *BGP update messages between neighboring ASes are delivered reliably in FIFO order, and have bounded delay;*
- A2. *Each AS sends out BGP update messages in bounded time after it updates its route profile;*
- A3. *Each BGP update message is processed immediately.*

**Proof:** Let the sequence of network route selections be  $\{r[t]\}_{t=0}^{\infty}$ .

Consider an arbitrary AS  $i$ . Let  $\mathcal{N}_i$  be the set of neighbors of  $i$ . For any  $j \in \mathcal{N}_i$ , let  $r_j[\tau_j^i(t)]$  be the latest route profile of  $j$  such that an update message has been sent to  $i$  with this route profile. Thus  $C_i(r_j[\tau_j^i(t)])$  is the set of paths in  $i$ 's routing cache learned from  $j$  at time  $t$ . The set of route profiles available to  $i$  is therefore  $A_i(\{r_j[\tau_j^i(t)] | j \in \mathcal{N}_i\})$ . Assumptions A1 and A2 imply that there exists  $t_d$  such that at any time  $t$ , for any neighbor  $j$  of  $i$ ,  $\tau_j^i(t) \geq t - t_d$ .

Although  $i$  may not know  $r_{-i}[t]$ , the payoff  $u_i(r_i, r_{-i})$  is only a function of  $r_i$ . (Recall that we consider only egress route selection in this case.) The BGP protocol, together with Assumption A3, implies that at any time  $t$

$$r_i[t] = \arg \max_{r_i \in A_i(\{r_j[\tau_j^i(t)] \mid j \in \mathcal{N}_i\})} u_i(r_i, r_{-i}[t]). \quad (8)$$

We shall prove the theorem by showing that  $t'' = t' + t_d$  satisfies Definition 2. In fact, for any  $t > t''$ , let  $H = \{r[s] \mid t' \leq s < t\}$ . For any neighbor  $j$  of  $i$ , we have  $\tau_j^i(t) \geq t - t_d \geq t'$ , thus  $r_j[\tau_j^i(t)] \in H_j$ . Therefore, there exists  $r_{-i} \in H_{-i}$  such that  $r_j[\tau_j^i(t)] = (r_{-i})_j$ . We shall show that  $r_i[t] \in U_i(H)$ . We have that  $r_i[t] \in A_i(r_{-i}) \subseteq A_i(H_{-i})$ . For any  $r'_i \in A_i(H_{-i})$ , if predicate P1 does not hold, then  $r'_i \in A_i(r_{-i})$ , which, together with Equation (8), implies that  $u_i(r_i[t], r_{-i}[t]) \geq u_i(r'_i, r_{-i}[t])$ . It follows that  $r_i[t] \in U_i(H)$ . ■

**Remark 1** *These three assumptions of the theorem should be valid under normal network operations. For example, when an AS applies route dampening, if the amount of time that a route is dampened has a finite upper bound, then the assumptions are still valid.*

**Remark 2** *Note that in Definition 2, AS  $i$  is not required to know the route selections  $r_{-i}[t]$  of the other ASes. AS  $i$  may not even know the sequence of times  $T$  and its set of all possible route profiles  $\mathcal{R}_i$ . In addition, the definition says nothing about the routing cache of  $i$ . The  $r_{-i} \in H_{-i}$  used in Definition 1 may have never appeared in  $i$ 's routing cache from time  $t'$  up to  $t$ . Moreover, at some time  $t$ ,  $r[t]$  may not even be consistent. All that is required is that the exhibited sequences of route selections  $r_i[t]$  and  $r[t]$  satisfy the requirement in the definition. The preceding theorem is an example clarifying this subtlety.*

### 6.3 A Sufficient Condition to Guarantee Convergence of Rational Route Selection Algorithms

Given the definition of rational route selection algorithms, in this section, we derive a sufficient condition to guarantee stability. The advantage of deriving a sufficient condition using the general notion of rational route selection algorithms is that we then only need to consider the asymptotic behaviors of route selection algorithms, allowing variations such as route dampening and limited route experimentation.

We first define the notion of stable route selection.

**Definition 3** *A network consisting of ASes each of which is running a rational route selection algorithm has a stable route selection, if the route selection of each AS has a single route profile, as time goes to infinite. Formally, the network has a stable route selection if  $\{r[t]\}_{t=0}^{\infty}$  converges.*

**Remark 3** *In the above definition, we require that, in a stable route selection, the route selection of each AS be a “pure” routing decision. We do not allow “mixed” strategies [43], since mixed strategies involve frequent route fluctuations, and are thus not desirable as “stable” solutions for global interdomain routing.*

We first observe the following important property of the operator  $U$ :

**Lemma 8** *The operator  $U$  is monotone: If  $P, Q \subseteq \mathcal{R}$  and  $P \subseteq Q$ , then  $U(P) \subseteq U(Q)$ .*

**Proof:** It suffices to show that  $U_i(P) \subseteq U_i(Q)$  for an arbitrary  $i$ .

Suppose  $r_i \in U_i(P)$ . We first notice that, since the operator  $A_i$  as defined in (4) is monotone,  $r_i \in A_i(P_{-i})$  implies  $r_i \in A_i(Q_{-i})$ . To prove  $r_i \in U_i(Q)$ , we only need to show that, for any  $r'_i \in A_i(Q_{-i})$ , at least one of the two predicates P1 and P2, which are defined in Definition 1, holds. We distinguish the following two cases:

1.  $r'_i \in A_i(P_{-i})$ . In this case, the fact that  $r_i \in U_i(P)$  implies that at least one of the two predicates P1 and P2 holds.
2.  $r'_i \notin A_i(P_{-i})$ . This case happens only if  $\forall r_{-i} \in P_{-i}, r'_i \notin A_i(r_{-i})$ . Thus predicate P1 holds in this case.

■

We now observe that sequences consistent with rational route selection share some common asymptotic properties:

**Theorem 9** *If  $\{r[t] | t \in T\}$  is consistent with rational route selection, then for each  $k$ , there exists  $t_k \in T$  such that, for all  $t \in T$  with  $t \geq t_k$ ,  $r[t] \in U^{(k)}(\mathcal{R})$ .*

**Proof:** For  $k = 0$ , the conclusion holds trivially (choosing  $t_0 = 0$ ) since for all  $t$ ,  $r[t] \in \mathcal{R} = U^{(0)}(\mathcal{R})$ .

Suppose the conclusion holds for  $k - 1$ . Then, there is a  $t_{k-1}$  such that for all  $t \geq t_{k-1}$ ,  $\{r[s] | t_{k-1} \leq s \leq t\} \subseteq U^{(k-1)}(\mathcal{R})$ . Since  $\{r[t] | t \in T\}$  is consistent with rational route selection, in Definition 2 we may choose  $t = t_{k-1}$  and we may take  $t_k > \max(t'', t_{k-1})$ . Therefore, for all  $t \geq t_k$ , we have that  $r[t] \in U(\{r[s] | t_{k-1} \leq s < t\}) \subseteq U(U^{(k-1)}(\mathcal{R})) = U^{(k)}(\mathcal{R})$ . ■

By Theorem 9, when the serially unoverwhelmed set  $U^\infty(\mathcal{R})$  is small, one can predict with precision the asymptotic behavior of a sequence of network route selections. In particular, if  $U^\infty(\mathcal{R})$  is a singleton, Theorem 9 immediately implies that the sequence will always converge to a unique network route selection. We therefore extend similar results in the context of strategic learning game [42] and learning in the Internet [20] to our route selection context.

**Proposition 10** *The network route selection of a network consisting of ASes running rational route selection algorithms asymptotically lie in the set  $U^\infty(\mathcal{R})$ . Thus, if  $U^\infty(\mathcal{R})$  is a singleton, the network is guaranteed the existence and uniqueness of stable route selection.*

One way to guarantee that  $U^\infty(\mathcal{R})$  is a singleton is the existence of a sequentially dominant route selection (SDRS). By a sequentially dominant route selection, we mean a partial order of the ASes, with the destination being the first one, such that given the route selection of the ASes before  $i$  in this partial order, the best route selection of  $i$  is determined, independent of the route selection of those after  $i$ . If a network has an SDRS, all routes other than the unique solution are not in the unoverwhelmed set. As such,  $U^\infty(\mathcal{R})$  is a singleton. The convergence of such networks

under any rational route selection algorithms, therefore, follows immediately from Theorem 7 and Proposition 10. Note that the existence of SDRS can be checked in polynomial time.

As an application of the preceding results, we derive a sufficient condition to guarantee routing convergence in a heterogeneous network where each AS runs any rational route selection algorithm, and its egress route selection satisfies the constraints imposed by business considerations [22].

**Theorem 11** *Assume a network where each AS runs any rational route selection algorithm, and selects egress routes independent of inbound traffic. Assume that 1) there is no provider-customer loop in the network; and 2) each AS adopts the typical export policy and the standard joint-route preference [55]. Then  $U^\infty(\mathcal{R})$  is a singleton; that is, the network is guaranteed to converge to the unique stable route.*

**Proof:** (*sketch*) When the conditions of the theorem are satisfied, we can use an induction proof to show the existence of an SDRS. Therefore, the network is guaranteed to converge to the unique stable route. ■

**Remark 4** *The preceding convergence result is more general than that proved in previous studies in that it is not limited to just homogeneous networks where each AS has to run the greedy, best-response BGP algorithm. Other actions, such as non-persistent experimentation, non-persistent dampening, are allowed.*

## 6.4 Instability of Networks under any Rational Route Selection Algorithms

Unfortunately, with inbound-dependency, there exist networks which have no stable route selection under any rational route selection algorithms; that is, we can arbitrarily assign route selection algorithm to each AS, so long each algorithm is a rational route selection algorithm, the network has no stable route selection.

In particular, Figure 14 is such an example network. Similar to the network in Figure 11, this network is constructed to satisfy all constraints imposed by AS business considerations; thus, if there were no inbound dependency, the network has a unique stable route selection [22]. Also similar to the network in Figure 11, this network does not appear to be a pathological case and can well happen in practice. Note that this network is a heterogeneous network, where the ranking of routes at  $S$  is inbound independent; while  $A$  and  $B$  are inbound dependent.

The instability of the example network in Figure 14 under any rational route selection scheme is established by the following result:

**Theorem 12** *Suppose that a sequence of network route selections  $\{r[t]\}_{t=0}^\infty$  is consistent with rational route selection and that it converges to a stable route selection  $r^*$ . Then the following holds for each AS  $i$ :*

$$\forall r'_i \in A_i(r^*_{-i}), u_i(r_i^*, r^*_{-i}) \geq u_i(r'_i, r^*_{-i}).$$

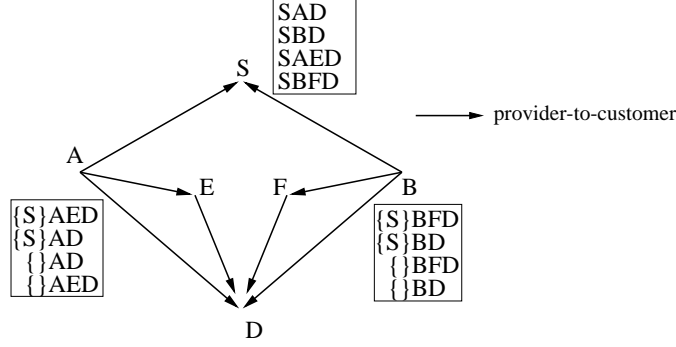


Figure 14: An example with instability.  $D$  is the only destination.

**Proof:** Since  $\{r[t]\}_{t=0}^{\infty}$  converges to  $r^*$ , there exists  $t'$  such that  $\forall t \geq t', r[t] = r^*$ . Since the sequence is consistent with rational route selection, there exists  $t'' > t'$ , such that  $\forall t > t''$  and  $\forall i, r_i[t] \in U_i(\{r[s] | t' \leq s < t\})$ . Notice that  $\{r[s] | t' \leq s < t\} = \{r^*\}$ , by definition of  $U_i$ , we have that

$$\forall r'_i \in A_i(r^*_{-i}), u_i(r_i^*, r^*_{-i}) \geq u_i(r'_i, r^*_{-i}).$$

■

An analysis of all of the possible network route selections of the example in Figure 14 shows that no network route selection satisfies the condition in Theorem 12. As a result, the network cannot converge to a stable route selection, under any rational route selection algorithm.

To further understand the example, consider the dynamics. When  $A$  and  $B$  choose  $AD$  and  $BFD$ . The outcome is  $SAD$  since  $S$  ranks  $SAD$  higher than  $SBFD$ . Then  $A$  has incentive to change from  $AD$  to  $AED$  since  $A$  ranks  $\{S\}AED$  higher than  $\{S\}AD$ . However,  $B$  realizes that, it can achieve a better outcome by changing  $BFD$  to  $BD$  since  $S$  will choose  $SBD$  over  $SAED$ . This in turn triggers  $A$  to switch from  $AED$  back to  $AD$ . Thus we end up with  $A$  chooses  $AD$  and  $B$  chooses  $BFD$  again, and the process continues forever.

## 7 Conclusions and Future Work

In this paper, we conduct the first systematic study on the stability and efficiency of using route selection to achieve interdomain traffic engineering objectives. We identify that interdomain traffic engineering requires that route selection be coordinated among multiple destinations and that coordinated route selection can introduce routing instability and inefficiency. We show the surprising result that the interaction of the routing of multiple destinations can cause routing instability even when the routing of each destination individually does have a unique solution. We propose a general, simple model to capture the fundamental feature of coordinated egress route selection behaviors for interdomain traffic engineering and construct P-graphs to derive a sufficient condition to guarantee convergence and existence of stable route selection. Taking into account constraints imposed by Internet business considerations, we show the pleasant but surprising result that egress route selection for interdomain traffic engineering in the current Internet is stable if there is no provider-customer loop, and all ASes follow the typical export policy and the standard joint-route

preference policy. We complement our analysis using simulations to investigate the likelihood of instability when the conditions are not satisfied. Our simulations based on realistic Internet AS topology show that if the policies are violated, even when a small number of ASes coordinate their routes for just two destinations, instability could happen.

Despite the success of the model and analysis of egress interdomain route selection, we also conduct the first systematic analysis on the stability of a more general model of interdomain route selection where an AS's ranking on routes depends on inbound traffic. We show that the common scheme of choosing the best routes according to the traffic-demand matrix of the preceding period could lead to instability, when the inbound traffic depends on route selection. We propose the notion of rational route selection algorithms, where inferior routes are iteratively eliminated. We derive a sufficient condition to check the stability of a network. We also show that there exist networks where routing will be unstable under any rational route selection algorithms, even when the ASes strictly follow the constraints imposed by AS business considerations.

There are many avenues for future work. In particular, although we show that the constraints imposed by Internet business considerations can guarantee convergence, ISPs may still have no incentives to follow these constraints. How to design incentive-compatible interdomain routing protocols which can guarantee convergence in the most generic setting is a major remaining challenge. The unstable network shown in Section 6.4 is particularly troubling in that it does not appear to be a pathological case, and thus could happen in practice. When we encounter such an unstable network setting in practice, there is still no satisfactory solution. Fundamentally, to stabilize the network, tradeoff between local optimality and global stability must be made. Thus, to design a stable route selection protocol, the ASes in a network must be willing to look into the future, form the right coalition, and sacrifice short-term benefits. Previous work such as route suppression (*e.g.*, [30]) and route dampening (*e.g.*, [41]) represents interesting potential directions. However, how to design interdomain routing protocols where the tradeoff between stability and local optimality is explicitly made in an incentive-compatible way is still a major remaining challenge.

## Acknowledgments

We thank Jiang Chen, Joan Feigenbaum, Eric Friedman, Arvind Krishnamurthy, Vijay Ramachandran, and Jennifer Rexford for giving us valuable comments. Our original proof of Theorem 4 was by induction; the one using Corollary 3 is pointed out by Aaron Jaggar. The proof also motivates us to define the notion of BGP subsystems. The connection between inbound-dependent route selection algorithms and traffic-demand-matrix-based algorithms is pointed out by Tim Griffin. We are grateful to their help. We also thank the network operators who replied to our surveys, and their replies helped us to identify the current trend in interdomain traffic engineering. RouteViews and Looking Glass servers make it possible for us to evaluate our schemes under realistic Internet topologies, and we are grateful to their providers.

## References

- [1] Mike Afegan and John Wroclawski. On the benefits and feasibility of incentive based routing infrastructure. In *Proceedings of ACM SIGCOMM '04 Workshop on Practice and Theory of Incentives and Game Theory in Networked Systems*, Portland, OR, September 2004.
- [2] Bengt Aspvall, Michael F Plass, and Robert Endre Tarjan. A linear-time algorithm for testing the truth of certain quantified boolean formulas. *Information Processing Letters*, 8(3):121–123, 1979.
- [3] Giuseppe Di Battista, Maurizio Patrignani, and Maurizio Pizzonia. Computing the types of the relationships between autonomous systems. In *Proceedings of IEEE INFOCOM '03*, San Francisco, CA, April 2003.
- [4] D. Bertsekas and R. Gallager. *Data Networks*. Prentice-Hall, Second Edition, 1992.
- [5] Thomas Bressoud, Rajeev Rastogi, and Mark Smith. Optimal configuration for BGP route selection. In *Proceedings of IEEE INFOCOM '03*, San Francisco, CA, April 2003.
- [6] CAIDA, Cooperative Association for Internet Data analysis. <http://www.caida.org/tools/>.
- [7] Alex Fabrikant, Christos H. Papadimitriou, and Kunal Talwar. On the complexity of pure equilibria. In *Proceedings of the 36th Annual Symposium on Theory of Computing*, Chicago, IL, 2004.
- [8] N. Feamster, J. Borkenhagen, and J. Rexford. Guidelines for interdomain traffic engineering. *ACM SIGCOMM Computer Communications Review*, October 2003.
- [9] N. Feamster and J. Rexford. Network-wide BGP route prediction for traffic engineering. In *Proceedings of ITCOM*, Boston, MA, August 2002.
- [10] Nick Feamster, Hari Balakrishnan, and Jennifer Rexford. Some foundational problems in interdomain routing. In *Proceedings of Third Workshop on Hot Topics in Networks (HotNets-III)*, San Diego, CA, November 2004.
- [11] Nick Feamster, Ramesh Johari, and Hari Balakrishnan. Stable policy routing with provider independence. In *Proceedings of ACM SIGCOMM '05*, August 2005. To appear.
- [12] Joan Feigenbaum, David Karger, Vahab Mirrokni, and Rahul Sami. Subjective-cost policy routing. Technical Report YALEU/DCS/TR-1302, Yale University, September 2004.
- [13] Joan Feigenbaum, Christos Papadimitriou, Rahul Sami, and Scott Shenker. A BGP-based mechanism for lowest-cost routing. In *Proceedings of the 21st ACM Symposium on Principles of Distributed Computing (PODC)*, pages 173–182, Monterey, CA, July 2002.

- [14] Joan Feigenbaum, Rahul Sami, and Scott Shenker. Mechanism design for policy routing. In *Proceedings of the 23rd ACM Symposium on Principles of Distributed Computing (PODC)*, pages 11–20, St. John’s, Newfoundland, Canada, July 2004.
- [15] Anja Feldmann, Olaf Maennel, Bruce Maggs, Nils Kammenhuber, Roberto De Prisco, and Ravi Sundaram. A methodology for estimating interdomain Web traffic demand. In *Proceedings of the Internet Measurement Conference*, October 2004.
- [16] Anja Feldmann, Olaf Maennel, Z. Morley Mao, Arthur Berger, and Bruce Maggs. Locating Internet routing instabilities. In *Proceedings of ACM SIGCOMM ’04*, Portland, OR, August 2004.
- [17] Anja Feldmann and Jennifer Rexford. IP network configuration for intradomain traffic engineering. *IEEE Network Magazine*, pages 46–57, Sept./Oct. 2001.
- [18] Bernard Fortz, Jennifer Rexford, and Mikkel Thorup. Traffic engineering with traditional IP routing protocols. *IEEE Communication Magazine*, October 2002.
- [19] Eric Friedman. Asynchronous learning in decentralized environments: A game theoretic approach. In K. Tumer and D. Wolpert, editors, *Collectives and the Design of Complex Systems*. Springer-Verlag, 2004.
- [20] Eric Friedman and Scott Shenker. Learning and implementation on the Internet. Working paper. Available at <http://www.orie.cornell.edu/~friedman/pfiles/decent.ps>, 1997.
- [21] Eric Friedman, Mikhael Shor, Scott Shenker, and Barry Sopher. An experiment on learning with limited information: Nonconvergence, experimentation cascades, and the advantage of being slow. *Games and Economic Behavior*, 47(2):325–352, 2004.
- [22] L. Gao and J. Rexford. Stable Internet routing without global coordination. *IEEE/ACM Transactions on Networking*, 9(6):681–692, December 2001.
- [23] Lixin Gao. On inferring autonomous system relationships in the Internet. *IEEE/ACM Transactions on Networking*, 9(6), December 2001.
- [24] Lixin Gao, Timothy G. Griffin, and Jennifer Rexford. Inherently safe backup routing with BGP. In *Proceedings of IEEE INFOCOM ’01*, Anchorage, AK, April 2001.
- [25] Ruomei Gao, Constantinos Dovrolis, and Ellen W. Zegura. Interdomain ingress traffic engineering through optimized AS-path prepending. In *Proceedings of Networking’05*, 2005.
- [26] David Goldenberg, Lili Qiu, Haiyong Xie, Yang Richard Yang, and Yin Zhang. Optimizing cost and performance for multihoming. In *Proceedings of ACM SIGCOMM ’04*, Portland, OR, August 2004.

- [27] R. Govindan and A. Reddy. An analysis of Internet inter-domain topology and route stability. In *Proceedings of IEEE INFOCOM '97*, Kobe, Japan, April 1997.
- [28] Timothy G. Griffin, Aaron D. Jaggard, and Vijay Ramachandran. Design principles of policy languages for path vector protocols. In *Proceedings of ACM SIGCOMM '03*, Karlsruhe, Germany, August 2003.
- [29] Timothy G. Griffin, F. Bruce Shepherd, and Gordon Wilfong. The stable paths problem and interdomain routing. *IEEE/ACM Transactions on Networking*, 10(22):232–243, April 2002.
- [30] Timothy G. Griffin and Gordon Wilfong. A safe path vector protocol. In *Proceedings of IEEE INFOCOM '00*, Tel Aviv, Israel, March 2000.
- [31] A.D. Jaggard and V. Ramachandran. Robustness of class-based path-vector systems. In *Proceedings of the 12nd International Conference on Network Protocols (ICNP) '04*, Berlin, Germany, October 2004.
- [32] A.D. Jaggard and V. Ramachandran. Relating two formal models of path-vector routing. In *Proceedings of IEEE INFOCOM '05*, Miami, FL, April 2005.
- [33] R. Johari and J. N. Tsitsiklis. Routing and peering in a competitive Internet. Available at: <http://web.mit.edu/jnt/www/publ.html>, January 2003.
- [34] C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian. Delayed Internet routing convergence. In *Proceedings of ACM SIGCOMM '00*, Stockholm, Sweden, August 2000.
- [35] C. Labovitz, G. R. Malan, and F. Jahanian. Internet routing instability. In *Proceedings of ACM SIGCOMM '97*, Cannes, France, September 1997.
- [36] Looking Glass servers. <http://www.traceroute.org>.
- [37] Nancy Lynch. *Distributed Algorithms*. Morgan Kaufmann Publishers, San Mateo, CA, 1996.
- [38] Ratul Mahajan, Maya Rodrig, David Wetherall, and John Zahorjan. Experiences applying game theory to system design. In *Proceedings of ACM SIGCOMM '04 Workshop on Practice and Theory of Incentives and Game Theory in Networked Systems*, Portland, OR, September 2004.
- [39] Ratul Mahajan, David Wetherall, and Thomas Anderson. Towards coordinated interdomain traffic engineering. In *Proceedings of Third Workshop on Hot Topics in Networks (HotNets-III)*, San Diego, CA, November 2004.
- [40] Ratul Mahajan, David Wetherall, and Thomas Anderson. Negotiation-based routing between neighboring domains. In *Proceedings of USENIX/ACM Symposium on Networked Systems Design and Implementation (NSDI '05)*, San Francisco, CA, May 2005.

- [41] Zhuoqing Morley Mao, Ramesh Govindan, George Varghese, and Randy H. Katz. Route flap damping exacerbates Internet routing convergence. *Computer Communication Review*, 32(4):221–233, 2002.
- [42] Paul Milgrom and John Roberts. Adaptive and sophisticated learning in normal form games. *Games and Economic Behaviors*, 3:82–100, 1991.
- [43] Martin J. Osborne and Ariel Rubenstein. *A Course in Game Theory*. The MIT Press, 1994.
- [44] B. Quoitin, S. Uhlig, and O. Bonaventure. Using redistribution communities for interdomain traffic engineering. In *QoFIS'02 LNCS 2511*, October 2002.
- [45] B. Quoitin, S. Uhlig, C. Pelsser, L. Swinnen, and O. Bonaventure. Interdomain traffic engineering with BGP. *IEEE Communications Magazine*, 41(5):122–128, May 2002.
- [46] RouteViews project. <http://www.routeviews.org/>.
- [47] Joao Luis Sobrinho. Network routing with path vector protocols: Theory and applications. In *Proceedings of ACM SIGCOMM '03*, Karlsruhe, Germany, August 2003.
- [48] L. Subramanian, S. Agarwal, J. Rexford, and R. H. Katz. Characterizing the Internet hierarchy from multiple vantage points. In *Proceedings of IEEE INFOCOM '02*, New York, NY, June 2002.
- [49] Lakshminarayanan Subramanian, Matthew Caesar, Cheng Tien Ee, Mark Handley, Z. Morley Mao, Scott Shenker, and Ion Stoica. Hlp: A next generation inter-domain routing protocol. In *Proceedings of ACM SIGCOMM '05*, August 2005. To appear.
- [50] R. Teixeira, T. Griffin, A. Shaikh, and G.M. Voelker. Network sensitivity to hot-potato disruptions. In *Proceedings of ACM SIGCOMM '04*, Portland, OR, August 2004.
- [51] Renata Teixeira, Aman Shaikh, Tim Griffin, and Jennifer Rexford. Dynamics of hot-potato routing in IP networks. In *Proceedings of Joint International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS)*, New York, NY, June 2004.
- [52] Steve Uhlig and Olivier Bonaventure. Implications of interdomain traffic characteristics on traffic engineering. In Jon Crowcroft and Anja Feldmann, editors, *Special issue on traffic engineering of European Transactions on Telecommunications*. 2002.
- [53] K. Varadhan, R. Govindan, and D. Estrin. Persistent route oscillations in inter-domain routing. *Computer Networks*, 32(1):1–16, 2000.
- [54] F. Wang and L. Gao. Inferring and characterizing Internet routing policies. In *Proceedings of the Internet Measurement Conference*, Miami, FL, October 2003.
- [55] Hao Wang, Haiyong Xie, Yang Richard Yang, Li Erran Li, Yanbin Liu, and Avi Silberschatz. On stable route selection for interdomain traffic engineering: Models, analysis, and guidelines. Technical Report YALEU/DCS/TR-1316, Yale University, February 2005.

- [56] Hui Wang, Rocky K.C. Chang, Dah-Ming Chiu, and John C.S. Lui. Characterizing the performance and stability issues of the AS path prepending method: Taxonomy, measurement study and analysis. In *Proceedings of ACM SIGCOMM Asia Workshop*, April 2005.

## A Notations

- $G$ : a simple, undirected graph representing network topology
- $V$ : the set of ASes in  $G$ ,  $V = \{1, \dots, N\}$
- $E$ : the set of interdomain links in  $G$
- $N$ : number of ASes in  $G$
- $\epsilon$ : the empty path
- $i, j, k, v_k$ : ASes in  $G$
- $v_k$ : an AS on a path in  $G$
- $P, Q, P_1, P_2$ : paths in  $G$
- $PQ$ : concatenation of paths  $P$  and  $Q$
- $P[v_i, v_j]$ : the subpath of  $P$  from  $v_i$  to  $v_j$
- $R$ : the set of all paths in  $G$
- $R_{i \rightarrow}$ : the set of paths originating from  $i$
- $R_{\rightarrow i}$ : the set of paths terminating at  $i$
- $R_{i \rightarrow j}$ : the set of paths from  $i$  to  $j$
- $\mathcal{S}$ : a set of source ASes
- $\mathcal{D}$ : a set of destination ASes
- $R_{\mathcal{S} \rightarrow \mathcal{D}}$ : the set of paths from any AS in  $\mathcal{S}$  to any AS in  $\mathcal{D}$
- $\mathcal{P}$ : a set of paths
- $\mathcal{P}_{\mathcal{S} \rightarrow \mathcal{D}}$ : the subset of paths of  $\mathcal{P}$  from any AS in  $\mathcal{S}$  to any AS in  $\mathcal{D}$
- $\text{export}(i, j, P)$ : export transformation at  $j$  on path  $P$  exported to  $i$
- $\text{import}(i, j, P)$ : import transformation at  $i$  on path  $P$  imported from  $j$
- $\text{pt}(i, j, P)$ : peering transformation on path  $P$  exported by  $j$  and imported by  $i$
- $\mathcal{D}_i$ : destinations of AS  $i$
- $\mathcal{D}_{ik}$ : disjoint subsets of  $\mathcal{D}_i$ , for  $k = 1, \dots, N_i$
- $N_i$ : number of disjoint subsets of  $\mathcal{D}_i$

- $r$ : a network route selection
- $\mathcal{R}$ : the set of all possible network route selections
- $r(i, j)$ : AS  $i$ 's selected route to AS  $j \in \mathcal{D}_i$
- $r_i$ : AS  $i$ 's route profile,  $r_i(j) = r(i, j)$
- $\mathcal{R}_i$ : the set of all possible route profiles for  $i$
- $r_i^{\mathcal{D}}$ : AS  $i$ 's partial route profile to destinations in  $\mathcal{D}$
- $\mathcal{R}_i^{\mathcal{D}}$ : the set of all possible partial route profiles from  $i$  to destinations in  $\mathcal{D}$
- $\mathcal{R}_i^{\mathcal{D}}(\mathcal{P})$ : the set of all possible partial route profiles for AS  $i$  with paths from  $i$  to destinations in  $\mathcal{D}$  drawn from  $\mathcal{P}$
- $r_{-i}$ : combined route selections of all ASes except  $i$
- $(r_{-i})_j$ : the route profile of AS  $j \neq i$  in  $r_{-i}$
- $\mathcal{R}_{-i}$ : the set of all possible combined route profiles of all ASes except  $i$
- $r_i^{\mathcal{D}_{ik}}, r_i^k$ : AS  $i$ 's partial route profile to destinations in  $\mathcal{D}_{ik}$
- $\sigma_i^{\mathcal{D}_{ik}}, \sigma_i^k$ : route selection function of AS  $i$  to destinations in  $\mathcal{D}_{ik}$
- $\lambda_i^{\mathcal{D}_{ik}}, \lambda_i^k$ : route ranking function of AS  $i$  for destinations in  $\mathcal{D}_{ik}$
- $\sigma_i$ : overall route selection function of AS  $i$
- $C_i$ : operator for available routes at AS  $i$
- $A_i$ : operator for available route profiles at AS  $i$
- $u_i(r_i, r_{-i})$ : traffic engineering utility function of AS  $i$
- $T$ : set of times,  $T = \{1, 2, \dots\}$
- $r[t]$ : network route selection at time  $t$
- $r_i[t]$ : route profile of AS  $i$  at time  $t$
- $H$ : a set of network route selections
- $H_i$ : the projection of  $H$  onto  $\mathcal{R}_i$ ,  $H_i = \{r_i \in \mathcal{R}_i | r \in H\}$ .
- $H_{-i}$ : the product of  $H_j$  for all  $j \neq i$ ,  $H_{-i} = \{r_{-i} \in \mathcal{R}_{-i} | (r_{-i})_j \in H_j, \forall j \neq i\}$ .
- $U_i$ : operator for unoverwhelmed route profiles of AS  $i$

- $U$ : operator for unoverwhelmed network route selections
- $\mathcal{N}_i$ : the set of neighbors of  $i$
- $\tau_j^i(t)$ :  $r_j[\tau_j^i(t)]$  is the latest route profile of  $j$  such that an update message has been sent to  $i$  with this route profile.
- $U^{(k)}$ : the  $k$ -th iterative application of operator  $U$
- $U^\infty$ : operator for serially unoverwhelmed set

## B An Alternative Proof of Theorem 4

**Proof:** Since there exists no provider-customer loop, we order all AS nodes into  $v_1, v_2, \dots, v_{|V|}$  such that each node appears before its provider(s). Let  $V_i = \{v_j | 1 \leq j \leq i\}, \forall 1 \leq i \leq |V|$ . Using strong induction on the order of a node, we now prove that each node has a unique set of customer routes to the destinations and BGP process converges to this set of routes. We refer to this as property  $P_c$ . We start the induction with node  $v_1$  since it has no customers; therefore, it has only one customer-reachable route, which is to itself. The customer routes to all other destinations are all empty. Note that we take a route from an AS to itself as a customer route of that AS. Thus, it is trivial that  $v_1$  satisfies  $P_c$ . We now assume that, property  $P_c$  holds for nodes  $v_1, \dots, v_k$ . We next consider node  $v_{k+1}$ . Denote by  $O(v_{k+1})$  all direct customers of node  $v_{k+1}$ . Note that  $O(v_{k+1}) \subseteq V_k$ . Note also that,  $O(v_{k+1})$  may be empty. In that case, we are done. Suppose  $O(v_{k+1})$  is not empty. By our induction hypothesis, each node  $u$  in  $V_k$  has a unique set of customer routes  $\hat{r}_u^C$ , and BGP process have converged to it. Because BGP update messages are reliable and the protocol is fair, node  $v_{k+1}$  will eventually receive and process the customer routes in  $\hat{r}_u^C$  from each customer  $u \in O(v_{k+1})$ . Note that  $v_{k+1}$ 's ranking function of customer routes  $\lambda_{v_{k+1}}^C$  only depends on the set of customer routes  $\{\hat{r}_u^C | u \in O(v_{k+1})\}$ , since by following condition 4,  $v_{k+1}$  selects routes for its customer-reachable destinations independent of routing decisions for peer-provider-reachable destinations. Therefore, node  $v_{k+1}$  will eventually pick  $\hat{r}_{v_{k+1}}^C$ , the best route selection for customer-reachable destinations.

We then prove that each node has a stable and unique set of peer and routes, and BGP process converges to it. We refer to this property as  $P_e$ . Therefore, each AS  $u$  eventually receives all routes to peer-reachable destinations from its peers. Given that  $u$  has already have  $\hat{r}_u^C$ ,

Finally, we prove that each node has a set of stable and unique peer and provider routes, and BGP converges to it. We refer to this property as  $P_{ep}$ . We order the set of nodes into  $v_1, v_2, \dots, v_{|V|}$  such that each node appears before its customers. We proceed by strong induction on the order of a node. According to the way we arrange the nodes, the node  $u$  with the lowest order does not have any providers. Then  $u$  has empty provider routes since we take the route from  $u$  to itself as a customer route of  $u$ . Also, if  $u$  has a peer link with  $v$ , then it will eventually receive  $v$ 's announced routes  $\hat{r}_v^C$ . Note that  $\hat{r}_v^C$  contains all routes from  $v$  to  $v$ 's customer-reachable destinations. Node  $u$  then extracts the route  $\hat{r}_v^C(d)$  in  $\hat{r}_v^C$  for each peer-reachable destination  $d \in D_u^E$  of node  $u$ , provided that  $v$  has a customer route to  $d$ . Note that, the fact that  $d$  is peer-reachable destination for  $u$  means that  $u$  does not have any customer route to  $d$  (by definition of peer-reachable destinations).  $u$  can determine its best route selection for peer-provider-reachable destinations using ranking function  $\lambda_u^{EP}$ . Thus property  $P_{ep}$  holds for  $u$ . Suppose that the property  $P_{ep}$  holds for all nodes  $V_k = \{v_i | 1 \leq i \leq k\}$  which have lower order than  $v_{k+1}$ . By our earlier proof, property  $P_c$  also hold for  $V_k$ . We now consider node  $v_{k+1}$ . Note that the providers of  $v_{k+1}$  are all in  $V_k$ . By our induction hypothesis and the property of BGP update process, node  $v_{k+1}$  will eventually receive the stable and unique routes from its neighboring providers for each provider-reachable destinations. Also, for any peer  $v$  of  $u$ , by our earlier proof, property  $P_c$  holds for  $v$ . Thus node  $v_{k+1}$  will eventually receive the stable and unique routes from its peers for each peer-reachable destinations. Because  $v_{k+1}$  has already had its stable and unique routes for its peer-reachable and provider-reachable

destinations,  $v_{k+1}$  can pick its best route selection using the ranking function  $\lambda_{v_{k+1}}^{EP}$ . Note that, the robustness property follows automatically because our proof does not make any assumption on the topology of the network. ■